(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
06 April 2023 (06.04.2023)

WIPO | PCT

(10) International Publication Number
WO 2023/055950 A1

(51) International Patent Classification:
G06F 21/10 (2013.01) *H04W 12/08* (2009.01)
H04L 9/40 (2022.01) *H04L 9/32* (2006.01)

(21) International Application Number:
PCT/US2022/045253

(22) International Filing Date:
29 September 2022 (29.09.2022)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/261,821 29 September 2021 (29.09.2021) US
63/291,929 20 December 2021 (20.12.2021) US
63/315,874 02 March 2022 (02.03.2022) US

(71) Applicant: **INTERTRUST TECHNOLOGIES CORPORATION** [US/US]; 400 N. McCarthy Blvd., Suite 220, Milpitas, California 95035 (US).

(72) Inventors: **NAGAO, Yutaka**; 437 Camille Circle, Unit 13, San Jose, California 95134 (US). **TIWARI, Vishisht**; 45060 Synergy Street, Apt. 272, Fremont, California 94538 (US). **KANNADKAR, Jayant**; 980 Henderson Avenue, Apt. 3, Sunnyvale, California 94086 (US). **ADVANI, Rohaan**; 403 Main Street, #504N, San Francisco, California 94105 (US).

(74) Agent: **DAVIS, John P.**; 26 S. Rio Grande Street, #2072, Salt Lake City, Utah 84101 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM,

(54) Title: CRYPTOGRAPHIC TOKEN RIGHTS MANAGEMENT SYSTEMS AND METHODS USING TRUSTED LEDGERS

(57) Abstract: The disclosure relates to, among other things, systems and methods for securely managing digital assets, products, and/or associated tokens in a dynamic way. Various embodiments disclosed herein may use trusted ledgers to securely record certain assertions relating to the digital assets, products, rightsholders, and/or various ecosystems participants. Such assertions may be used in connection with rights binding, content packaging, management, and/or protection operations in connection with a variety of transactions associated with digital assets, products, and/or associated tokens.

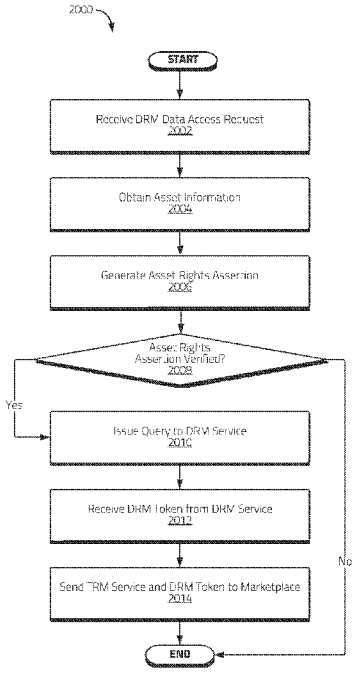


Figure 20

[Continued on next page]

WO 2023/055950 A1

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

CRYPTOGRAPHIC TOKEN RIGHTS MANAGEMENT SYSTEMS AND METHODS USING TRUSTED LEDGERS

RELATED APPLICATIONS

[0001] This application claims the benefit of priority under 35 U.S.C. §119(e) to U.S. Provisional Patent Application No. 63/261,821 filed September 29, 2021, and entitled "CRYPTOPGRAHIC TOKEN MANAGEMENT SYSTEMS AND METHODS," to U.S. Provisional Patent Application No. 63/291,929 filed December 20, 2021, and entitled "SYSTEMS AND METHODS FOR CRYPTOGRAPHIC TOKEN MANAGEMENT USING TRUSTED LEDGERS," and to U.S. Provisional Patent Application No. 63/315,874 filed March 2, 2022, and entitled "CRYPTOGRAPHIC TOKEN RIGHTS MANAGEMENT SYSTEMS AND METHODS USING TRUSTED LEDGERS," the contents of which are hereby incorporated by referenced in their entireties.

COPYRIGHT AUTHORIZATION

[0002] Portions of the disclosure of this patent document may contain material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the U.S. Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

SUMMARY

[0003] The present disclosure relates generally to systems and methods for managing cryptographic tokens. More specifically, the present disclosure relates to systems and methods for managing non-fungible tokens ("NFT") and rights to associated content in a NFT marketplace using trusted ledgers.

[0004] An NFT is a type of cryptographic token on a trusted distributed ledger (*e.g.*, a blockchain ledger) that may represent an asset, which may comprise a digital asset and/or

a physical asset. NFTs may be used in a variety of contexts, with applications varying from simple digital asset transactions to complex collateral loans.

[0005] NFT paradigms may be implemented using a variety of architectures. For example, a centralized server may be used to store metadata associated with a digital asset. The server may create a reference to that data and may store a reference link in a trusted ledger such as a cryptographic blockchain. This approach may be relatively high speed and inexpensive in implement in terms of costs and processing power, and may allow for flexible updating of referenced metadata which is not directly stored in the trusted lodger. However, if the server goes down, there may not be enough data in the trusted ledger to allow one to obtain the original asset, potentially rendering the NFT essentially useless. In addition, if an owner deletes or changes the location of the data and the asset is not stored on the central server, then the link to the asset may not function as intended.

[0006] Another approach to implementing an NFT paradigm may involve the storage of metadata on a trusted ledger itself. This may allow relevant data to be stored on the ledger with less third-party dependence and more reliability in the event an NFT marketplace goes down. There are, however, additional costs for this type of implementation in terms of expense, speed, and/or processing demands.

[0007] Embodiments of the systems and methods may provide an improved architecture for managing digital assets. In certain embodiments, metadata associated with digital assets may be stored in a server database, and the correlation of digital assets and associated metadata may be securely stored in a trusted ledger, which may comprise a blockchain ledger.

[0008] Many conventional NFT marketplaces may not be well suited to address copyright concerns. For example, if a marketplace customer purchases a NFT for a song, then the customer can download the song and easily upload it on other platforms, potentially in violation of other's rights to the song. Embodiments of the disclosed systems and methods may use digital rights management ("DRM") technology in

connection with NFT management paradigms to help ameliorate at least some of these concerns. For example, using various aspects of the disclosed systems and methods, a customer can listen to a purchased song, but they may not be able to download the song and upload it on other platforms in violation of their rights to the song, because the purchased song may be protected with DRM techniques.

[0009] In various disclosed embodiments, trusted databases, ledgers, and/or the like (which may be generally referred to herein as "trusted ledgers" and/or variations of the same), may be used to record and/or otherwise manage various assertions associated with digital assets and/or NFTs. In some embodiments, such databases and/or ledgers may be distributed, and may be referred to herein as trusted immutable distributed assertion ledgers ("TIDALs") and/or variations of the same.

[0010] In some embodiments, trusted ledgers used in connection with various aspects of the disclosed embodiments may comprise blockchain ledgers. Databases and/or ledgers may, in various embodiments, be public, private, and/or a combination thereof. In certain embodiments, a TIDAL may comprise a public indelible distributed database ("PIDD"). TIDALs consistent with various aspects of the disclosed embodiments may be associated with a variety of properties including, for example, ledger processes that may be resistant to byzantine failures, entries that may be immutable and/or relatively immutable, entries that may be time-synced (at least in part), entries that may be scalable, and/or entries that may be available for relatively fast lookup.

[0011] Embodiments of the disclosed systems and methods may allow for rights binding, packaging, management, and/or protection operations in connection with a variety of NFT transactions (*e.g.*, minting NFTs, transferring NFTs, etc.). DRM functionality and/or key management may be provided, facilitating management of digital assets, products, and/or NFTs based on authenticated access rights.

[0012] In some embodiments, digital assets (*e.g.*, digital content) may be packaged into products, serialized, and/or managed in accordance with one or more enforced business conditions. Business conditions may include permitted actions that may be

performed in connection with a digital asset and/or an associated product and may be set by entities that hold certain rights to the digital asset and/or associated product. For example and without limitation, business conditions may be used to manage digital assets and/or associated NFTs based on sell models, rental models, subscription models, rent-ownership models, and/or the like.

[0013] Embodiments of the disclosed systems and methods may provide elements of an ecosystem for registering and uploading of digital assets, downloading of digital assets and/or products, creation and/or packaging of digital assets and/or products, listing of digital assets and/or products, delisting of digital assets and/or products, updating of digital assets and/or products, sharing of digital assets and/or products, playback of digital assets and/or products, and/or the like. Further embodiments may facilitate compensation of a rights holder and/or plurality of rights holders associated with a digital asset, product, and/or associated NFT(s) based on consumer usage and/or interaction with associated digital assets, products, and/or NFTs. Certain embodiments may use one or more commercial blockchain ledgers such as, for example, the FLOW blockchain, and blockchain connector services to provide mechanisms to readily implement aspects of the disclosed systems and methods with such established blockchain architectures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The inventive body of work will be readily understood by referring to the following detailed description in conjunction with the accompanying drawings, in which:

[0015] **Figure 1A** illustrates a first part of a non-limiting example of a process for registering an NFT asset with a trusted service consistent with certain embodiments disclosed herein.

[0016] **Figure 1B** illustrates a second part of a non-limiting example of a process for registering an NFT asset with a trusted service consistent with certain embodiments disclosed herein.

[0017] **Figure 2A** illustrates a first part of a non-limiting example of a process for purchasing and accessing an NFT asset using a trusted service consistent with certain embodiments disclosed herein.

[0018] **Figure 2B** illustrates a second part of a non-limiting example of a process for purchasing and accessing an NFT asset using a trusted service consistent with certain embodiments disclosed herein.

[0019] **Figure 2C** illustrates a third part of a non-limiting example of a process for purchasing and accessing an NFT asset using a trusted service consistent with certain embodiments disclosed herein.

[0020] **Figure 3** illustrates a non-limiting example of aspects of a NFT transaction process consistent with certain embodiments disclosed herein.

[0021] **Figure 4A** illustrates a first part of a non-limiting example of an asset upload process consistent with certain embodiments disclosed herein

[0022] **Figure 4B** illustrates a second part of a non-limiting example of an asset upload process consistent with certain embodiments disclosed herein

[0023] **Figure 5A** illustrates a first part of a non-limiting example of a product creation process consistent with certain embodiments disclosed herein

[0024] **Figure 5B** illustrates a second part of a non-limiting example of a product creation process consistent with certain embodiments disclosed herein

[0025] **Figure 5C** illustrates a third part of a non-limiting example of a product creation process consistent with certain embodiments disclosed herein

[0026] **Figure 6A** illustrates a first part of a non-limiting example of a download process consistent with certain embodiments disclosed herein.

[0027] **Figure 6B** illustrates a second part of a non-limiting example of a download process consistent with certain embodiments disclosed herein.

[0028] **Figure 6C** illustrates a third part of a non-limiting example of a download process consistent with certain embodiments disclosed herein.

[0029] **Figure 6D** illustrates a fourth part of a non-limiting example of a download process consistent with certain embodiments disclosed herein.

[0030] **Figure 7A** illustrates a first part of a non-limiting example of an asset playback process consistent with certain embodiments disclosed herein.

[0031] **Figure 7B** illustrates a second part of a non-limiting example of an asset playback process consistent with certain embodiments disclosed herein.

[0032] **Figure 7C** illustrates a third part of a non-limiting example of an asset playback process consistent with certain embodiments disclosed herein.

[0033] **Figure 8A** illustrates a first part of a non-limiting example of an asset update process consistent with certain embodiments disclosed herein.

[0034] **Figure 8B** illustrates a second part of a non-limiting example of an asset update process consistent with certain embodiments disclosed herein.

[0035] **Figure 8C** illustrates a third part of a non-limiting example of an asset update process consistent with certain embodiments disclosed herein.

[0036] **Figure 9A** illustrates a first part of a non-limiting example of a product update process consistent with certain embodiments disclosed herein.

[0037] **Figure 9B** illustrates a second part of a non-limiting example of a product update process consistent with certain embodiments disclosed herein.

[0038] **Figure 9C** illustrates a third part of a non-limiting example of a product update process consistent with certain embodiments disclosed herein.

[0039] **Figure 10A** illustrates a first part of a non-limiting example of a product delisting process consistent with certain embodiments disclosed herein.

[0040] **Figure 10B** illustrates a second part of a non-limiting example of a product delist process consistent with certain embodiments disclosed herein.

[0041] **Figure 11A** illustrates a first part of a non-limiting example of an asset delete process consistent with certain embodiments disclosed herein.

[0042] **Figure 11B** illustrates a second part of a non-limiting example of an asset delete process consistent with certain embodiments disclosed herein.

[0043] **Figure 12A** illustrates a first part of a non-limiting example of an asset unbundling process consistent with certain embodiments disclosed herein.

[0044] **Figure 12B** illustrates a second part of a non-limiting example of an asset unbundling process consistent with certain embodiments disclosed herein.

[0045] **Figure 13A** illustrates a first part of a non-limiting example of a product invite process consistent with certain embodiments disclosed herein.

[0046] **Figure 13B** illustrates a second part of a non-limiting example of a product invite process consistent with certain embodiments disclosed herein.

[0047] **Figure 13C** illustrates a third part of a non-limiting example of a product invite process consistent with certain embodiments disclosed herein.

[0048] **Figure 14A** illustrates a first part of a non-limiting example of a product disinvite process consistent with certain embodiments disclosed herein.

[0049] **Figure 14B** illustrates a second part of a non-limiting example of a product disinvite process consistent with certain embodiments disclosed herein.

[0050] **Figure 15A** illustrates a first part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0051] **Figure 15B** illustrates a second part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0052] **Figure 15C** illustrates a third part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0053] **Figure 15D** illustrates a fourth part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0054] **Figure 15E** illustrates a fifth part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0055] **Figure 15F** illustrates a sixth part of a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0056] **Figure 16A** illustrates a first part of a non-limiting example of an asset update process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0057] **Figure 16B** illustrates a second part of a non-limiting example of an asset update process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0058] **Figure 16C** illustrates a third part of a non-limiting example of an asset update process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0059] **Figure 17A** illustrates a first part of a non-limiting example of an asset purchase process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0060] **Figure 17B** illustrates a second part of a non-limiting example of an asset purchase process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0061] **Figure 17C** illustrates a third part of a non-limiting example of an asset purchase process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0062] **Figure 18A** illustrates a first part of a non-limiting example of an asset rental process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0063] **Figure 18B** illustrates a second part of a non-limiting example of an asset rental process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0064] **Figure 18C** illustrates a third part of a non-limiting example of an asset rental process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0065] **Figure 19A** illustrates a first part of a non-limiting example of an asset playback process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0066] **Figure 19B** illustrates a second part of a non-limiting example of an asset playback process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0067] **Figure 19C** illustrates a third part of a non-limiting example of an asset playback process using a blockchain ledger consistent with certain embodiments disclosed herein.

[0068] **Figure 20** illustrates a flow chart of a non-limiting example of a method of managing a digital asset consistent with certain embodiments of the present disclosure.

[0069] **Figure 21** illustrates a non-limiting example of a system that may be used to implement certain embodiments of the systems and methods of the present disclosure.

DETAILED DESCRIPTION

[0070] A description of systems and methods consistent with embodiments of the present disclosure is provided herein. While several embodiments are described, it should be understood that the disclosure is not limited to any one embodiment, but instead encompasses numerous alternatives, modifications, and equivalents. In addition, while numerous specific details are set forth in the following description in order to provide a thorough understanding of the embodiments disclosed herein, some embodiments can be practiced without some or all of these details. Moreover, for the purpose of clarity, certain technical material that is known in the related art has not been described in detail in order to avoid unnecessarily obscuring the disclosure.

[0071] The embodiments of the disclosure may be understood by reference to the drawings, where in certain instances, but not necessarily all instances, like parts may be designated by like numerals or descriptions. The components of the disclosed embodiments, as generally described and/or illustrated in the figures herein, could be arranged and designed in a wide variety of different configurations. Thus, the following description of the embodiments of the systems and methods of the disclosure is not intended to limit the scope of the disclosure, but is merely representative of possible embodiments of the disclosure. In addition, the steps of any method disclosed herein do not necessarily need to be executed in any specific order, or even sequentially, nor need the steps be executed only once, unless otherwise specified.

[0072] Embodiments of the disclosed systems and methods may provide for dynamic management of digital assets, products, and/or associated NFTs in a secure way, where the rights of rights holders in such assets, products, and/or NFTs are respected and/or otherwise enforced. Various embodiments disclosed herein may use trusted ledgers to securely record certain assertions relating to the digital assets, products, rightsholders, and/or various ecosystems participants. Such assertions may be used in connection with

rights binding, content packaging, management, and/or protection operations in connection with a variety of transactions associated with digital assets, products, and/or associated NFTs.

[0073] Certain embodiments disclosed herein may be used to manage digital assets, products, and/or associated NFTs in accordance with a variety of associated business conditions, supporting sell models, rental models, subscription models, rent-ownership models, and/or the like, as well as supporting a variety of rights holder compensation models. DRM and/or key management services may be used in connection with managing access to digital assets and/or associated products, which may comprise and/or otherwise be associated with digital content such as, for example and without limitation, audio content, video content, image content, text content, and/or the like. Further embodiments facilitate integration of various aspects of the disclosed systems and methods with established commercial blockchain and digital wallet services using blockchain connector services.

[0074] Certain aspects of the disclosed processes may be performed by and/or otherwise be performed in connection with one or more entities, services, and/or systems that may include, for example and without limitation, one or more of:

[0075] • Marketplace service – An entity, which may comprise a third-party entity that uses a trusted data management platform ("TIP") service (or, in certain instances herein, more generally a "trusted data management service") for NFT and/or DRM implementations.

[0076] • Owner – A digital asset owner that may upload content to the marketplace service.

[0077] • TIDAL – A distributed blockchain service and/or other suitable trusted ledger service that may be provided by the TIP service and/or another service and/or entity. Although various examples described herein may use a TIDAL, it will be appreciated that various embodiments and aspects of the disclosed systems and methods may be implemented using a variety of other trusted databases, ledgers, and/or

blockchains that may, or may not, have certain attributes associated with a TIDAL. As used herein, a TIDAL may generally describe an assertion ledger having certain properties of a TIDAL as detailed herein (and/or a subset thereof), and in some instances may describe a TIDAL with entries derived and/or otherwise generated based on entries in one or more other ledgers and/or TIDALs, which in some instances may be referred to alternatively and/or additionally as a derivative TIDAL.

[0078] • DB – A database that may be created by the marketplace service that may interact with data virtualization services.

[0079] • File storage – A file storage service such as, for example and without limitation, Amazon S3 cloud, that may be used to store an encrypted digital asset.

[0080] • TIP – In various illustrative examples included herein, "TIP" may reference an API offered by a TIP service for the marketplace service to access various services that may include, for example and without limitation, one or more of the TIDAL, DB, and/or file storage services.

[0081] • Consumer – A buyer of digital assets through the marketplace service.

[0082] • Payment gateway – A service facilitating payment between various services and entities in connection with certain transactions involving digital assets.

[0083] **NFT Asset Registration and Upload**

[0084] **Figures 1A-1B** illustrate a non-limiting example of a process for registering an NFT asset consistent with various embodiments of the present disclosure. A variety of entities, services, and/or systems participating in the ecosystem may engage in various aspects of the illustrated processes including, for example and without limitation, an owner 100 of an NFT and/or a digital asset, a marketplace service 102, a TIP service 104, a TIDAL 106, a DB 108 managed by a database service and/or the TIP service 104, a file storage service 110, a payment gateway 112, and/or a consumer 114. Various functionality of the systems, services, entities, and/or execution environments illustrated

in **Figures 1A-1B** and elsewhere herein may be integrated into and/or otherwise performed by single systems and/or services and/or any suitable combination of multiple systems and/or services.

[0085] It will be appreciated that various aspects of the disclosed embodiments may be used in connection with a variety of types of digital assets, which may be represented by and/or otherwise associated with NFTs. For example, various disclosed embodiments may be used in managing audio, video, image, and/or written content and/or associated NFTs generated by and/or otherwise associated with a variety of content creators, distributors, managers, and/or other parties have interest and/or rights to content within a content ecosystem.

[0086] Various aspects of the disclosed NFT management processes are described below in reference to a number of constituent steps.

[0087] **Account Initialization**

[0088] A marketplace service 102 that would like to interact with and/or otherwise use a TIP service 104 may contact the TIP administrator to be provisioned with a TIP account. Once the account is set up, the marketplace administrator may connect their database to the TIP service 104. In some embodiments, the TIP administrator may create a dataset using data virtualization ("DV") services that points to a DB 108 managed by a database service and/or the TIP service 104.

[0089] The marketplace service 102 can have a user interface that may allow owners 100 and/or consumers 114 to sign up for services. In certain embodiments, this interaction may not necessarily create an account directly but may notify a marketplace administrator to request that an owner 100 and/or consumer 114 log in using a suitable user authentication method (*e.g.* Google login and/or any other suitable login protocol). Once the account is created and their e-mail verified, the owners 100 and/or consumers 114 may log in to the marketplace service 100 using their associated user authentication credentials. After logging into the marketplace service 100, the owner 100 and/or

consumer 114 may be redirected to a user account interface. An owner 100 and/or consumer 114 can enter payment receiving details on this page.

[0090] Upload Phase

[0091] Steps 1.1 – 1.2:

[0092] The marketplace service 102 may comprise a third-party service using TIP services 104. The TIP service 104 may offer secure execution environment ("SEE") deployment services including, for example and without limitation:

[0093] • Secure communication and key management for a TIDAL 106 by an SEE.

[0094] • Controlling access to the DB 108 by multiple entities (*e.g.*, owner 100, consumer 114, etc.) with DV and/or identity access management ("IAM") services.

[0095] • Secure management of digital asset encryption and key management by an SEE.

[0096] An owner 100 may login to the marketplace service 102 using their associated authentication credentials (*e.g.*, using a Google account via Open ID connect and/or another authentication protocol). The marketplace service 102 may authenticate the owner 100, potentially returning an indication of success if the owner 100 is authenticated.

[0097] Step 1.3:

[0098] The owner 100 may upload a digital asset (*e.g.*, image, video, audio, text, etc. and/or combinations thereof) to the marketplace service 102. In some embodiments, the owner 100 may provide metadata with the uploaded digital asset. The metadata may comprise, for example and without limitation, one or more of:

[0099] • A title.

[00100] • A description.

[00101] • One or more business conditions, which in some embodiments may comprise public and/or private conditions. For example, a public business condition may allow a consumer (*e.g.*, consumer 114) to access (*e.g.*, watch and/or listen to) a digital asset without engaging in an outright transaction (*e.g.*, buy, rent, subscription). Public business conditions in this non-limiting example may comprise, for example, sellout and associated price information. A private business condition may allow, for example, a consumer (e.g., consumer 114) to access (*e.g.*, watch and/or listen to) a digital asset after having completed a purchasing transaction (*e.g.*, buy, rent, subscription). Private business conditions in this non-limiting example may comprise, for example, sellout and associated price information, rental and associated price information, and/or subscription and associated price information, and/or the like.

[00102] It will be appreciated that a variety of business conditions may be associated with digital assets and/or products via metadata in connection with various embodiments disclosed herein. Business conditions may express one or more granular conditions and/or permissions including, for example and without limitation, one or more of:

[00103] • Do not allow for playback under certain conditions. For example and without limitation, playback for a digital asset and/or product may be restricted before a date and/or time and/or after a date and/or time (*e.g.,* a playback period), within a certain country, location, and/or jurisdiction and/or combinations thereof.

[00104] • Do not allow for rental/subscription to a digital asset and/or product under certain conditions. For example and without limitation, rental of and/or subscription to a digital asset and/or product may be restricted before a date and/or time and/or after a date and/or time (*e.g.,* a rental period), within a certain country, location, and/or jurisdiction, for less than a minimum price, for more than a maximum price, and/or combinations thereof.

[00105] • Do not allow for sale to a digital asset and/or product under certain conditions. For example and without limitation, sale of a digital asset and/or product may be restricted before a date and/or time and/or after a date and/or time (*e.g.*, a sale period),

within a certain country, location, and/or jurisdiction, for less than a minimum price, for more than a maximum price, and/or combinations thereof.

[00106] • Do not allow for inviting a friend under certain conditions. For example and without limitation, invite permissions may be restricted before a date and/or time and/or after a date and/or time (*e.g.*, an invite period), within a certain country, location, and/or jurisdiction, not to exceed a maximum quantity of invitations (*e.g*, simultaneous and/or total), and/or combinations thereof.

[00107] • Limit the maximum quantity of rentals (*e.g.*, simultaneous and/or total rentals)

[00108] • Limiting the maximum number of sales allowed.

[00109] Business conditions articulated in metadata may further express an indication of fractional ownership and/or other rights held in connection with digital assets and/or associated products and/or NFTs. In certain embodiments, such fractional rights may be expressed in terms of ownership and/or rights holding shares, cuts, and/or percentages, although other methods and/or ways of implementing fractional ownership and/or rights holding are also envisioned. For example and without limitation, business conditions may express an indication of a name and/or other identity of rights holders and associated ownership rights and/or percentages of ownership rights for a digital asset, product, and/or associated NFT(s). In various embodiments, such fractional ownership and/or rights and/or rights may be traded and/or otherwise sold and/or exchanged using services enabled by embodiments of the disclosed ecosystem. Moreover, as discussed in more detail below, payment models and/or mechanisms between owners and/or rightsholders may be implemented based, at least in part, on fractional ownership and/or rights information detailed in business conditions.

[00110] **Step 1.4:**

[00111] The marketplace service 102 may add identity information associated with the owner 100 to a payload that may include the digital asset and may send the payload along

with the associated metadata to the TIP service 104 (*e.g.*, via a REST API). In some embodiments, the marketplace service 102 may receive and/or otherwise retrieve information stored in a user account associated with the owner 100 such as, for example, a name, address, email, contact information, payment receiving details, and/or the like, and may add this information to the metadata.

[00112] Steps 1.5-1.7:

[00113] The TIP service 104 may generate assertions, which may comprise a base assertion and/or account assertion, based, at least in part, on the uploaded digital asset. For example, in some embodiments, an assertion may comprise a hash generated based on the digital asset and/or associated information (*e.g.*, marketplace and/or owner account information) and/or one or more hashes generated based on the same. In some embodiments, an assertion may comprise a signed hash value, allowing for a measure of authentication of a recorded assertion.

[00114] Assertions may be recorded in a trusted ledger, which may comprise a TIDAL 106, although other types of ledgers may also be used in connection with various disclosed embodiments. As detailed above, the base assertion may be generated based on the digital asset by, for example and without limitation, hashing the digital asset. The account assertion may be generated based on the base assertion and identity information associated with the owner 100. For example, the account assertion may comprise a hash of the base assertion concatenated with a token and/or other identification information associated with the owner 100 of the digital asset.

[00115] The TIP service 104 may check whether the base assertion (*e.g.*, the hash) exists in the TIDAL 106 and/or a derivative ledger (*e.g.*, a ledger, which may be a TIDAL, with entries derived and/or otherwise generated based on entries in one or more other ledgers and/or TIDALs). When the hash for the base assertion does not exist in the TIDAL 106 (*e.g.*, based on an indication that no duplicates exist in the ledger), the TIP service 104 may ingest and/or otherwise store the base assertion and the account assertion in the TIDAL 106. In some embodiments, the base assertion and the account assertion

may be cryptographically signed with a key securely associated with the TIP service 104. The TIDAL 106 and/or a service managing the TIDAL 106 may be managed in accordance with an ingestion policy configured to accept entries into the TIDAL 106 when an entry to be recorded in the TIDAL 106 is signed by the TIP service 104 and/or another authorized service and/or entity.

[00116] In some embodiments, when the hash for the base assertion already exists in the TIDAL 106, the TIP service 104 may check whether the account assertion also exists in the TIDAL 106. If the account assertion already exists in the TIDAL 106, this may indicate that the original owner of the digital asset has attempted to re-upload the digital asset. In this case, the process may proceed to Step 1.8. If the account assertion does not exist in the TIDAL 106, this may indicate that a different owner tried to upload a copied digital asset to the marketplace. In this case, an alert message may be returned to terminate further transactions.

[00117] Step 1.9-1.14:

[00118] Once the base assertion and/or account assertion checks are completed, the TIP service 104 may store market metadata for the digital assets in the DB 108. The TIP service 104 may also create a market assertion and ingest the generated market assertion into the TIDAL 106. In some embodiments, the market assertion may be generated based on the account assertion and the market metadata (which in some embodiments may be defined in Step 1.3). For example, the market assertion may comprise a signed hash of a concatenation of the account assertion and the market metadata. In certain implementations, market assertions may be cryptographically signed with a key associated with the TIP service 104, and the TIDAL's ingestion policy may be configured to accept entries only when they are signed by the TIP service 104.

[00119] To protect the uploaded digital asset, the TIP service 104 may generate a content key, which may comprise a symmetric cryptographic content key. The TIP service 104 may encrypt the digital asset with the generated content key, and then encrypt the content key with a key associated with the service (*e.g.*, a secret symmetric key, a

public key associated with a private key securely stored by the TIP service, etc.). The TIP service secret key may, in some embodiments, be stored in a secure storage and/or vault of the SEE. The TIP service 104 may store the encrypted content key in the DB 108 and the encrypted digital asset in the file storage service 110. After successful completion, the TIP service 104 may return the market assertion to the marketplace service 102. In some embodiments, an indication of the successful upload and/or registration of a digital asset may be provided to the owner 100 by the marketplace service 102.

[00120] Download Phase

[00121] **Figures 2A-2B** illustrate a non-limiting example of a process for purchasing and accessing an NFT asset consistent with certain embodiments disclosed herein.

[00122] Steps 2.1-2.2:

[00123] A consumer 114 may login to the marketplace service 102 using their associated authentication credentials (*e.g.*, using a Google account via Open ID connect and/or another authentication protocol). The marketplace service 102 may authenticate the consumer 114 based on the provided credentials and/or may return an indication of a successful authentication (or unsuccessful if the credentials are not authenticated).

[00124] Steps 2.3-2.8:

[00125] The marketplace service 102 may request the TIP service 104 to retrieve verified assets. In some embodiments, the TIP service 104 may obtain asset details from the DB 108 and may generate market assertions. The TIP service 104 may check market assertions against a TIDAL 106 to verify the latest status and/or integrity. The TIP service 104 may send back verified asset details to the marketplace service 102. The marketplace service 102 may then display market metadata (which may be defined by the owner 100 as detailed above) associated with verified digital assets to the consumer 114. The consumer 114 may then browse and purchase available digital assets from the marketplace service 102. Depending on business conditions for the digital asset, a

consumers may, for example and without limitation, purchase (*e.g.*, own), rent, and/or subscribe to the digital asset.

[00126] Steps 2.9-2.14:

[00127] A variety of consumer transactions may be supported by the marketplace service 102 including, for example and without limitation, purchase, rental, playback, and/or subscription transactions, although other types of transactions associated with digital assets are also contemplated. For example, when a consumer 114 requests to access a digital asset for watching and/or listening (*e.g.*, playback), the process may proceed to Step 3.1 of **Figure 2C**. In another example, when a consumer 114 requests to purchase public or private digital assets for sellout, rent, and subscription (*e.g.*, Step 2.9), the marketplace service 102 may redirect the consumer 114 to a secure payment gateway service 112. In some embodiments, the secure payment gateway service 112 may comprise a third-party secure payment service such as, for example and without limitation, PayPal, although other payment services may also be used.

[00128] For example, as illustrated in **Figure 2B**, the marketplace service 102 may call an API of the payment gateway service 112. The payment gateway service 112 may respond to the marketplace service 102 with one or more available payment options. The consumer 114 may select a suitable payment method supported by the marketplace service 102 such as, for example and without limitation, credit and/or debit card payment, PayPal, and/or the like. The consumer 114 may confirm payment details with the marketplace service 102.

[00129] The marketplace service 102 may call an API of the payment gateway service 112 with the requested order information and may receive an indication from the payment gateway service 112 indicating successful payment by the consumer 114. After successful payment, the marketplace service 102 may receive the amount from the consumer 114. The marketplace service 102 may then distribute the payment to respective owner 100 in accordance with one or more applicable business agreements (as discussed below in connection with **Figure 3**).

[00130] Step 2.15:

[00131] The marketplace service 102 may create a signed JSON web token ("JWT") with the consumer's identity, agreed business condition(s), base assertion, and/or other transaction details. The marketplace service 102 may send the generated JWT to the TIP service 104. The marketplace service 102 may use its private key to sign the JWT.

[00132] Steps 2.16-2.18:

[00133] The TIP service 104 may verify the JWT signature using a public key associated with the marketplace service 102. For sale of the public and/or private digital assets, the TIP service 104 may create a new account assertion based on the base assertion and the new owner's identity (*e.g.*, an identity of the consumer 114). For example, in some embodiments, the TIP service 104 may create a new account assertion by generating a cryptographic hash based on the base assertion and information associated with the new owner's identity (*e.g.*, a hash calculated on a concatenation of the base assertion and the new owner's identity information). The TIP service 104 may register the new account assertion to the TIDAL 106. The TIP service 104 may further register any old account assertion to the TIDAL 106 and/or a derivative of the same as false and/or otherwise outdated.

[00134] For rent and/or subscription of private digital assets, the TIP service 104 may generate a rights assertion based on the account assertion and agreed business condition information. For example, in some embodiments, the TIP service 104 may create a new rights assertion by generating a cryptographic hash based on the account assertion and information associated with the agreed business condition information (*e.g.*, a hash calculated on a concatenation of the account assertion and the agreed business condition information). The agreed business condition information may include, for example and without limitation, rental and/or subscription detail information and/or the identity information associated with the consumer 114. The rights assertion may be signed by the TIP service 104 (*e.g.*, cryptographically signed by a private key securely associated with the TIP service 104).

[00135] The TIP service 104 may register the rights assertion in the TIDAL. The TIDAL's ingestion policy may be configured to accept assertions including signatures made by the TIP service 104. Based on the purchase transaction, the TIP service 104 may also store the transaction info in the DB 108 by correlating it to the purchased digital assets. This transaction information may comprise, for example and without limitation, an identity associated with the consumer 114 and the agreed business condition information. The TIP service 104 may use the rights assertion for verification purposes at Step 3.5, shown in connection with **Figure 2C**, for rent of and/or subscription to private digital assets.

[00136] **Steps 2.19-2.21:**

[00137] For accessing public or private digital assets for watching, listening, reading, and/or the like, the TIP service 104 may generate a unique link to access the digital asset. This link may comprise the base assertion and may map to the original digital asset on an associated file storage service 110. The TIP service 104 may provide the marketplace service 102 with the generated link. A consumer 114 may then be provided the link by the marketplace service 102 for use in accessing the digital asset. For example and without limitation, in the case of rights managed audio and/or video content, the consumer 114 can streaming the content using the link. In certain embodiments, when a consumer 114 access the content, a DRM token may be generated (*e.g.*, as may be the case in connection with audio, video, and/or other DRM managed content).

[00138] **Steps 3.1-3.8:**

[00139] When consumer 114 opens the link to access the digital asset, a request may be sent to the marketplace service 102. The marketplace service 102 may send a verification request to the TIP service 104. In some embodiments, the request may comprise a signed JWT with the base assertion and an identity of the consumer 114.

[00140] If the digital asset is marked as public in the market assertion and verified the TIDAL 106, the TIP service 104 may retrieve the base assertion from the JWT and the

encrypted content key from the digital asset from the DB 108 by using the retrieved base assertion.

[00141] If the digital asset is marked as private in the market assertion and/or verified with TIDAL 106, the TIP service 104 may retrieve the base assertion and the identity information of the consumer 114 from the JWT. The agreed business condition and/or identity of the owner 100 may be retrieved from the DB 108 using the retrieved base assertion and consumer identity information. The TIP service 104 may generate a rights assertion and may check the rights assertion against the TIDAL 106 (which may be a derivative TIDAL). If the rights assertion exists in the TIDAL 106 (*e.g.*, based on the TIDAL returning an indication that the assertion has been verified), then the TIP service 104 may check the content of the agreed business condition (*e.g.*, check whether a rental period expires or not). If the agreed business condition is valid for the consumer 114, the TIP service 104 may retrieve the encrypted content key for the digital asset from the DB 108 using the retrieved base assertion.

[00142] In some embodiments where DRM methodologies are be applied in connection with various aspects of the disclosed systems and methods (for example, in connection with audio and/or video content), the TIP service 104 may decrypt the encrypted content key with its secret key. The TIP service 104 may then call the DRM service APIs to issue a token for a DRM license by providing the base assertion (*e.g.*, content ID), the content key, and/or agreed business condition information (*e.g.*, rental period, output control, etc.). Once a token has been issued by the DRM service, the TIP service 104 may return the token to the marketplace service 102. The marketplace service 102 may then initiate the playback of the digital asset.

[00143] In further embodiments, the TIP service 104 may decrypt the encrypted content key with its secret key. The TIP service 104 may then retrieve the encrypted digital asset from the file storage and decrypt it with the content key. The TIP service 104 may then make the decrypted digital asset to be available to the consumer 114 using a link. The consumer 114 may then access the digital asset via the link provided by the marketplace service 102.

[00144] Transaction Clearing Phase

[00145] **Figure 3** illustrates a non-limiting example of aspects of a NFT transaction process consistent with certain embodiments disclosed herein. As discussed in more detail below, in various embodiments, the marketplace service 102 may distribute payment to respective owners 100 in accordance with one or more applicable business agreements and/or conditions.

[00146] Steps 4.1-4.5:

[00147] The marketplace service 102 may request details of new transactions to the TIP service 104. For example, the marketplace service 102 may request details of new transactions to the TIP service 104 on a periodic basis (*e.g.*, weekly). Based on the purchase amount, agreed business condition information, and/or associated marketplace service fees (which may be retrieved by the TIP service 104 from the DB 108), the marketplace service 102 may generate an amount payable to the owner 100 (or owners) of the digital assets. The marketplace service 102 may make payment to the owner 100 using available payment receiving details.

[00148] Token Rights Management Service Ecosystem

[00149] Consistent with embodiments disclosed herein, a token rights management ("TRM") service may be used in connection with managing NFTs and/or other digital assets. As discussed above, an NFT may comprise a cryptographic token recorded on a trusted ledger, such as a blockchain ledger, that may represent a digital asset. Application of NFTs may be broader, however, varying from simple media transactions to complex collateral loans. **Figure 4A-14B** illustrate various conceptual diagrams showing various aspects of a process for managing NFTs using a TRM service in a TRM ecosystem consistent with certain embodiments disclosed herein.

[00150] TRM Ecosystem Entities, Services, and/or Systems

[00151] Certain aspects of the disclosed processes using TRM services may be performed by and/or otherwise in connection with one or more entities, services, and/or systems, which may be similar to the entities, services, and/or systems described above in connection with **Figures 1A-Figure 3**. These entities, services, and/or systems may include, for example and without limitation, one or more of:

[00152] • Marketplace services 102 – An entity, which may comprise a third-party entity that uses TRM services 402 to implement various NFT management processes. In various embodiments, a TRM service 402 may trust the marketplace service 102 for TRM API calls. A marketplace service 102 may manipulate data for their own marketplace, but may not readily be able to, or in some implementations may be unable to manipulate data of another marketplace.

[00153] • Marketplace administrator – An administrator that may set up and/or otherwise configure a marketplace service 102 with TRM APIs and TIP services 104. In some implementations, a TIP account ID of a marketplace administrator may be used as a marketplace ID.

[00154] • Original Rights Holder ("ORH") 400 – An ORH 400, which may be a content creator, distributor, and/or any other service, entity, and/or party that may hold original rights in a digital asset and may upload digital assets and/or create products.

[00155] • Owner 100 – A product owner 100, who may have created their own product (*e.g.*, as an ORH 400) and/or may have purchased it through the marketplace service 102.

[00156] • TIDAL 106 – A distributed blockchain service and/or other suitable trusted ledger service that may be provided by the TIP and/or another service and/or entity. Although various examples described herein may use a TIDAL, it will be appreciated that various embodiments and aspects of the disclosed systems and methods may be implemented using a variety of other trusted databases and/or ledgers that may, or may not, have certain attributes associated with a TIDAL detailed herein. Moreover, although various examples are described herein in connection with a TIDAL, it will be

appreciated that a derivative TIDAL may also be used in connection with various aspects of the disclosed embodiments.

[00157] • DB 108 – A database that may be created and/or otherwise managed by the marketplace service 102 and/or the TIP service 104 that may interact with data virtualization services.

[00158] • File storage service 110 – A file storage service 110 such as, for example and without limitation, Amazon S3 cloud, that may be used to store an encrypted digital asset.

[00159] • TRM services 402 – TRM services 402 may be implemented via an API for the marketplace service to access TIDAL 106, DB 108, and/or file storage services 110.

[00160] • TIP service 104 – The TIP service 104 used for various TRM implementations that may provide governance among different marketplace services 102. TIP services 104 may include for example and without limitation, one or more of: (1) TIP DV, which may comprise DV services of a TIP service 104 used for fetching digital assets, products, and/or transactions using an appropriate query (*e.g.*, A SQL query); (2) TIP IAM services which may comprise IAM services of a TIP used for governance among different marketplace services 102; (3) TIP vault, which may comprise a TIP service 104 for securely storing protected data – Key Encryption Keys ("KEKs") stored in a TIP vault may be accessible to only TRM administrators in certain implementations (i.e., marketplace administrators may have limited access (if any)); and (4) TIP SEE, which may comprise a TIP service 104 that provides an SEE for running a TRM backend that, in some implementations, may be used for communication between TRM service 402, TIDAL 106, DB 108, and/or file storage services.

[00161] • Consumer 114 – A buyer of digital assets through the marketplace service.

[00162] • Payment Gateway 112 – A payment gateway service 112 may comprise a service facilitating payment between various services and entities in connection with certain transactions involving digital assets.

[00163] • TRM administrator – An administrator that sets up marketplace services 102 TRM access including TIP account creation.

[00164] • Key management services ("KMS") 114 – A service configured to provide certain key management functionality including, for example and without limitation, key generation, content key storage and/or management, and/or the like. KMSs 104 may in some embodiments be offered by and/or included in a DRM services.

[00165] • Invitee – A user invited by a product owner 100 to consume an owner's 100 product.

[00166] TRM Ecosystem Identifiers

[00167] Various embodiments disclosed herein may use a variety of identifiers in connection with digital assets, users, entities, services, and/or other aspects of the disclosed systems and methods. These identifiers may comprise, for example and without limitation, one or more of:

[00168] • User ID – Users (*e.g.*, owner 100, consumer 114, etc.) may be assigned a user ID and/or an associated identifier. User IDs may comprise, for example and without limitation, one or more of: (1) an owner ID, which may comprise a user ID of the owner 100 of a product; (2) an ORH ID, which may comprise a user ID of the ORH 400 of a product (*e.g.*, a product creator), (3) a consumer ID, which may comprise a user ID of the consumer 114 of product (*e.g.*, a user who purchases and/or rents a product); and an invitee ID, which may comprise the user ID for an invitee.

[00169] • Asset ID – Assets may be assigned an asset ID after successful upload.

[00170] • Product ID – When a product is created, a product ID may be generated (*e.g.*, a "UUID") and may be assigned to copies of the new product.

[00171] • Task ID – Tasks may be assigned a task ID.

[00172] • Product serial ID – Each product copy may be assigned a unique identifier represented by a product serial ID.

[00173] • Transaction ID – Transactions may be assigned a transaction ID. These transactions may include, for example and without limitation, buy, sellout, subscription, and/or other transactions that may be represented by business conditions.

[00174] • Key ID ("KID") – A key ID may be generated as a key identifier for a KMS 404.

[00175] • Stat ID – A stat ID may correspond to a stat row in a stats table.

[00176] • Marketplace ID – A marketplace ID may correspond to a marketplace service 102. In some embodiments, a marketplace ID may be the same as an account ID of a marketplace service 102 on a TIP service 104.

[00177] **TRM Ecosystem Tables**

[00178] A variety of tables may be used in connection with managing NFTs in connection with various disclosed embodiments. These tables may comprise, for example and without limitation, one or more of:

[00179] • Task table – A task table may comprise, for example and without limitation, one or more of a task ID, marketplace ID, creator ID, asset ID/product ID (potentially used for update), title, task timestamp, asset/product, status (*e.g.*, processing, done, failed, deleted, etc.), a type of task (*e.g.*, upload, update, delist, delete), etc.

[00180] • Asset table – An asset table may comprise, for example and without limitation, one or more of an asset ID (which may be unique for each asset), marketplace ID, media type (*e.g.*, audio, video, image, text, etc.), encrypted content key, KID, file location URL, title, asset assertion, fact (*e.g.*, a hash of a file content), upload timestamp, ORH ID, metadata(JSON), initial supply, remaining supply, preview URL, etc.

[00181] • Product table – A product table may comprise, for example and without limitation, one or more of a product ID, product serial ID, marketplace ID, name, assets (*e.g.*, list of asset IDs included in bundle), serial number, type (*e.g.*, public or private), preview URL, listed timestamp, owner ID, ORH ID, business-condition (*e.g.*, an array of business conditions), sell price, rent price, rental period, metadata (*e.g.*, JSON), product assertion, fact, delist, delete, invitees (*e.g.*, list of invitee IDs), etc.

[00182] • Transaction table – A transaction table may comprise, for example and without limitation, one or more of a transaction ID, marketplace ID, owner ID, consumer ID, product serial ID, timestamp, business condition, sell-price, rent-price, rental expiry date, asset ID, etc.

[00183] • Marketplace user table – A marketplace user table may comprise, for example and without limitation, one or more of a user ID, e-mail, name, account creation timestamp, balance, role (*e.g.*, a role supported by marketplace services 102), etc.

[00184] • Marketplace table – A marketplace table may comprise, for example and without limitation, one or more of a marketplace ID, name, contact e-mail, and/or the like. This table may be managed manually by a TRM administrator. When a new marketplace service 102 is provisioned, its TIP account may be created and an entry in this table may be created with marketplace ID set to the newly created TIP account ID.

[00185] TRM Ecosystem Physical Datasets

[00186] Physical datasets in embodiments of the disclosed TRM ecosystem may comprise TIP datasets that map to the TRM Tables described above. These datasets may be used to fetch assets, products, transactions from TIP using queries (*e.g.*, directly from TIP using SQL queries).

[00187] A variety of datasets may be used in connection with managing NFTs in connection with various disclosed embodiments. These datasets may comprise, for example and without limitation, one or more of:

[00188] • Task physical dataset – A TIP dataset that may map to the task table. The marketplace administrator may have view and/or query privileges on this dataset. The TRM administrator may have administrator privileges on this dataset.

[00189] • Asset physical dataset – A TIP dataset that may map to the asset table. The marketplace administrator may have view and/or query privileges on this dataset. The TRM administrator may have administrator privileges on this dataset.

[00190] • Product physical dataset – A TIP dataset that may map to the product table. The marketplace administrator may have view and/or query privileges on this dataset. The TRM administrator may have administrator privileges on this dataset.

[00191] • Transaction physical dataset – A TIP dataset that may map to the transaction table. The marketplace administrator may have view and/or query privileges on this dataset. The TRM administrator may have administrator privileges on this dataset.

[00192] • Marketplace user physical dataset – A TIP dataset that may map to the marketplace user table. This dataset may be created by each marketplace administrator and may be supported by a marketplace. The marketplace administrator may have administrator privileges on this dataset. The TRM administrator may have no privileges on this dataset.

[00193] TRM Ecosystem Asset Assertions

[00194] A variety of assertions may be used in connection with managing NFTs in connection with various disclosed embodiments. These assertions may comprise asset assertions and product assertions. Asset assertions may comprise, for example and without limitation, one or more of:

[00195] • Asset assertion – An asset assertion may comprise a hash of file content, which may be used in some embodiments to detect copied content using a TIDAL 106. In some embodiments, an assert assertion may comprise a hash of a hash of file content,

which may comprise a binary of an audio, video, and/or image file (*e.g.*, hash(hash(file content)).

[00196] • Asset ownership assertion – An asset ownership assertion may prove ownership of the asset and/or enable playback by the ORH 400. In some embodiments, the asset ownership assertion may comprise an asset assertion + ORH ID ("+" used here and elsewhere herein may denote a concatenation of values). In further embodiments, the asset ownership assertion serves as the proof of ownership of an asset via a TIDAL 106.

[00197] TRM Ecosystem Product Assertions

[00198] A variety of assertions may be used in connection with managing NFTs in connection with various disclosed embodiments. These assertions may comprise, for example and without limitation, one or more of:

[00199] • Product Assertion – A product assertion may, in some embodiments, serve as a base for the subsequent assertions. Product assertions may be stored in a product table for each entry for quick assertion computations. In some embodiments, a product assertion may comprise a hash of a concatenation of asset assertions of individual assets and a serial number. Asset assertions may in some implementations be concatenated in alphabetical order.

[00200] • Product ownership assertion – The product ownership assertion may serve as a proof of product ownership in a TIDAL 106. In some embodiments, the product ownership assertion may comprise a product assertion and an owner ID.

[00201] • Market assertion – A market assertion may validate metadata in a DB 108. A market assertion may serve as a verification of product market metadata of product table entries in a TIDAL 106. A market assertion may comprise a product assertion and market metadata. In at least one non-limiting example, the market metadata may be determined according to: market metadata = product metadata (*e.g.*, title + description) + business metadata (*e.g.*, business condition + sell price + rent price + rental period).

[00202] • Rights assertion – A rights assertion may validate transactions. The rights assertion may serve as a verification of a transaction date in a TIDAL 106. In at least one non-limiting example, the rights assertion may be calculated according to: rights assertion = product assertion + transaction price (*e.g.*, amount paid by consumer 114) + business condition + consumer ID.

[00203] • Product invite assertion – A product invite assertion may validate an invitee's access to a product. In at least one non-limiting example, the product invite assertion may be calculated according to: product invite assertion = product ownership assertion + invitee ID.

[00204] TRM Ecosystem APIs

[00205] A variety of APIs may be used in connection with managing NFTs in connection with various disclosed embodiments. These APIs may comprise, for example and without limitation a TRM verification API of a TRM service 402. This API may allow the marketplace service 102 to verify data using a TIDAL 106. In some embodiments, the input may comprise a list of asset IDs/product serial IDs/transaction IDs and the API may verify if the data on database corresponding to these IDs is valid. For assets, verification may use asset ownership assertions using a TIDAL 106; for products, verification may use product ownership assertions and market assertions using a TIDAL 106; for transactions, verification may use rights assertions using a TIDAL 106. In some embodiments, the API may accept as input one or more of a list of asset IDs, product serial IDs, and/or transaction IDs and/or type of IDs (*e.g.*, asset, product, transaction, etc.). In certain embodiments, the API may return a list of IDs that failed the verification. A returned empty array may mean data for all IDs were valid.

[00206] TRM Ecosystem Business Conditions

[00207] A variety of business conditions may be used in connection with managing NFTs in connection with various disclosed embodiments. These business conditions may comprise, for example and without limitation, one or more of:

[00208] • Sellout – The ownership of a product is transferred from one user to another. Associated business metadata may comprise, for example and without limitation, price information.

[00209] • Rent – A user can access a product for a limited amount of time. The ownership may remain with the current owner. Associated business metadata may comprise, for example and without limitation, price and/or rental periods.

[00210] • Subscription – A User can access a product if they pay the subscription fees. The ownership may remain with the current owner. Associated business metadata may comprise, for example and without limitation, price and/or rental period.

[00211] • Rent-Ownership – The ownership may be transferred from ORH 400 to another user for a limited amount of time. A new owner can make this product available for rent-ownership/rent/subscription. However, after the rent-ownership time period, the ownership may revert back to the ORH 400. Associated business metadata may comprise price and/or rental period. A new owner in some implementations may not set rental period (for rent or subscription) beyond the original rental period (for rent-ownership) set by the ORH 400. When a product is sold, a scheduled job may be created. This job may transfer the ownership back to ORH 400 after the rent-ownership rental period. The scheduled job may be a thread in a program that runs after the rental period expires. In some embodiments, changing the time period for the rent-ownership may be restricted to the ORH 400. Subsequent rent-ownership transactions from new owners may have the original rental-ownership period. The business condition for rent-ownership may, in some embodiments, be restricted to the ORH 400. In some implementations, the business condition may not be changed to rent-ownership if existing rent/subscription transactions exist for the product.

[00212] **TRM Ecosystem Metadata**

[00213] A variety of metadata definitions may be used in connection with managing NFTs in connection with various disclosed embodiments. These metadata definitions may comprise, for example and without limitation, one or more of:

[00214] • Asset metadata – Asset metadata may in some embodiments comprise a title.

[00215] • Product metadata – Product metadata may in some embodiments comprise one or more of a title, a description, and/or any additional metadata decided by the marketplace administrator.

[00216] • Business Metadata – Business metadata may in some embodiments comprise one or more business conditions, sell prices, rent prices, rent periods, subscription prices, subscription periods, rent-ownership prices, and/or rent-ownership periods.

[00217] Delisting a Product

[00218] In various embodiments, any product may at any point be delisted from the marketplace service 102 by the owner, which may be the ORH 400. This means that the product may no longer be available on the marketplace service 102 for consumers 114 to purchase, rent, and/or subscribe. In some embodiments, product table entries may have a field called "delist" which may take on the values "0" or "1". A delist value of "0", signifying false, may mean that the product is currently available on the marketplace service 102. On the other hand, a delist value of "1", indicating true, may mean that the product is currently delisted and so not currently available on the marketplace service 102.

[00219] Asset Supply and Product Copies

[00220] When a digital asset is uploaded to the marketplace service 102, the ORH 400 may indicate its initial supply. In some implementations, once selected, the initial supply of the asset may not be changed by the ORH 400 (although this may not be the case in every implementation). A digital asset's initial supply may set an upper limit on the number of times the asset may be bundled in different products, so that the ORH 400 may not indefinitely add the digital asset to new products.

[00221] When a product is created, the ORH 400 may select the number of copies of the new product to mint. Each copy may have a unique product serial ID, but a uniform product ID to correlate the different copies. Each such copy may be referred to as a product copy. In at least one non-limiting example, suppose a new Product A is created with 5 copies- in the backend, this may result in 5 new entries in the Product table, say A0 (Serial #0), A1 (Serial #1), A2 (Serial #2), A3 (Serial #3), A4 (Serial #4). Each entry, or product copy, may have a unique product serial ID, the UUID that serves as the primary key of the product table. However, the product copies may have the same product ID, a UUID that is used to correlate the copies of Product A.

[00222] In some embodiments, when a digital asset is bundled into a product, its remaining supply may decrease by the total copies of the Product. For example and without limitation, consider Asset A with an initial supply of 100. The ORH 400 may decide to create Product X, including Asset A, with 10 copies. The remaining supply of A may now be 90. Now, the ORH 400 may create another Product Y, which bundles A, and the ORH 400 may decide there will be 50 copies of Product Y. The remaining supply of A may now be reduced to 40. If the ORH 400 decides to delete Product X, the remaining supply of A may increase by the number of unsold copies of X (*e.g.*, remaining supply of A will become 50 if Product X remains unsold entirely). In some embodiments, the number of versions of a product may be limited by the minimum remaining supply across the bundled assets in that product.

[00223] **Marketplace Preparations**

[00224] In some embodiments, a marketplace service 102 that would like to use a TRM service 402 consistent with various aspects of the disclosed embodiments may contact the TRM administrator to be provisioned a TIP account. The TRM administrator 402 may create an API key and client, which may be used by the marketplace service 102 for obtaining access tokens to authenticate the TRM API calls.

[00225] The asset, product, and transaction tables may be common for different marketplace services 102 and may be under the TRM TIP instance. Marketplaces may

query these tables via TIP DV services, as the dataset restrictions may be set up by the TRM administrator such that each marketplace service 102 may view those entries matching their marketplace ID (*e.g.*, TIP account ID). In some embodiments, the marketplace service 102 may manage their own user authentication and marketplace user table.

[00226] In some embodiments, the marketplace can have an interface that allows ORH 400, owners 100, and/or consumers 114 to 'sign up', which may not create an account directly but may notify the marketplace administrator to let the ORH 400, owners 100, and/or consumers 114 log in using a user authentication (*e.g.*, Google login). Once the Account is created and their email verified, the ORH 400, owners 100, and/or consumers 114 may log in to the marketplace service 102 using their user authentication credentials. After login to the marketplace service 102, the ORH 400, owners 100, and/or consumers 114 may be redirected to a user account page. The ORH 400, owners 100, and/or consumers 114 can enter payment receiving details on this page.

[00227] Asset Upload

[00228] **Figures 4A-4B** illustrate a non-limiting example of an asset upload process consistent with certain embodiments disclosed herein

[00229] Steps 1.1-1.2: User Login

[00230] In some embodiments, the ORH 400 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the ORH 400, issuing the ORH 400 an authentication token if successfully authenticated.

[00231] Steps 1.3-1.4: Upload Asset

[00232] The ORH 400 may upload the digital asset (*e.g.*, image, video, text, audio, etc.) to the marketplace service 102. The ORH 400 may specify the initial supply for the

digital asset, which may an upper limit on the number of times the asset may be bundled in different products. The ORH 400 may upload a preview file to serve as a thumbnail for the asset on the marketplace portal. The ORH 400 may also provide asset title, description and any other metadata, which may be determined and/or otherwise set by the marketplace service 102.

[00233] The marketplace service 102 may obtain a TIP access token from TIP IAM services using its API key and secret. The marketplace services may provide this access token along with the request data (*e.g.*, digital asset file, preview file, initial supply, metadata) to the TRM API of the TRM service 402.

[00234] Steps 1.5-1.7: Verify Marketplace

[00235] The marketplace TIP access token may be verified via the TIP IAM service. The marketplace ID may be obtained from the IAM service and/or user information endpoint. The marketplace service 102 may then be verified by interacting with the DB 108 (*e.g.*, it may be determined that the marketplace ID exists in the marketplace table in the DB 108, as the marketplace table stores the list of verified marketplace services 102).

[00236] Steps 1.8-1.11: Asset Assertions

[00237] The TRM service 402 may generate the asset assertion. In some embodiments, the asset assertion may comprise a hash of the file content of the digital asset. An asset ownership assertion may also be generated by the TRM service 402. In certain embodiments, the asset ownership assertion may be generated by concatenating the ORH ID to the asset assertion.

[00238] The TRM service 402 may query a TIDAL 106 to check if the asset assertion already exists - if it does, it may indicate that the digital asset has already been uploaded to the TRM service 402, and so the copied content may be rejected. If not, the new asset and asset ownership assertions may be inserted into the TIDAL 106.

[00239] Steps 1.12-1.20: Asset Encryption

[00240] A task entry may be created in the task table with status "processing" while the digital asset encryption takes place. In various embodiments, one or more steps of an asset encryption process may be performed as an asynchronous task. To protect the digital asset, the KEK may be retrieved from the TIP vault and/or a new KEK may be generated and stored in the TIP vault if it does not exist already. The KID may be generated. The KEK and KID may be sent to a KMS 404 where a content key – K – may be generated and returned to TRM service 402. The digital asset file may be encrypted using the content key using K. An encrypted digital asset and/or preview may be stored in the file storage service 110, and an associated location (*e.g.*, URL) that may be used to access the encrypted digital asset and/or preview may be returned to the TRM service 402. The entry may be created in the asset table including the asset information. The entry may be updated at the task table with status set to "done" and the asset ID may also be updated in the task table.

[00241] In some embodiments, to protect the digital asset, the secret key may be obtained from TIP vault. A new secret key may be generated and stored in the TIP vault if it does not already exist. In some embodiments, the content key may be encrypted using the secret key. The entry may be created at the asset table including the associated asset information. The entry may be updated in the task table with status set to "done" and the asset ID may also be updated in the task table.

[00242] In some embodiments, an indication of the successful uploading of the digital asset may be communicated to the marketplace service 102 and/or the ORH 400 from the TRM 402.

[00243] **Product Creation**

[00244] **Figures 5A-5C** illustrates a non-limiting example of a product creation process consistent with certain embodiments disclosed herein

[00245] **Step 2.1-2.2: User Login**

[00246] In some embodiments, the ORH 400 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the ORH 400, issuing the ORH 400 an authentication token if successfully authenticated.

[00247] **Step 2.3-2.9: Create Product**

[00248] Once the ORH 400 decides to create a new product, the marketplace service 102 may fetch the list of assets created by the ORH 400 from the asset physical dataset via a DV offered by the TIP service 104 using its TIP access token. The ORH 400 may use the TRM verification API to validate if the fetched assets are valid. Once assets are verified, the ORH 400 may decide which of their assets to bundle in this new product, and provide the supporting product information that may include, for example and without limitation, one or more of:

[00249] • Number of copies to mint.

[00250] • Title.

[00251] • Preview file (*e.g.*, thumbnail, etc).

[00252] • Product type – In some embodiments a product type may be public or private. Public products may be accessed by any consumers (*e.g.* rendered for viewing and/or listening, etc.). Private products may be accessed by consumers after a purchasing transaction.

[00253] • Business condition – Business conditions may comprise, for example and without limitation, sellout, rent, subscription, rent-ownership, and/or the like.

[00254] • Business metadata – Business metadata may, in certain embodiments, comprise price (*e.g.*, for sellout, rental, subscription, rent-ownership) and/or rental period (*e.g.*, for rental, subscription, rent-ownership, etc.).

[00255] • Product description and/or any other metadata, which may be defined by the marketplace service 102.

[00256] The marketplace service 102 may generate and issue a product creation request to an API of the TRM service. In some embodiments, the product creation request may comprise the marketplace TIP access token.

[00257] Steps 2.10-2.12: Verify Marketplace

[00258] The marketplace TIP access token included in the product creation request may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00259] Steps 2.13-2.14: Verify Ownership of Underlying Assets

[00260] For each asset bundled in the product, the TRM service 402 may generate the asset ownership assertion and may verify it with the TIDAL 106 (*e.g.*, checking whether the assertion exists in the TIDAL 106). If any of the asset ownership assertions are unverified, the product creation request may be rejected, as the ownership of underlying assets may not be verified.

[00261] Steps 2.15-2.16: Check Asset Supply Counts

[00262] The remaining supply counts for underlying assets in the product may be retrieved from the asset table. If the remaining supply counts are less than the specified total copies of the product, the product creation request may be rejected. In such a case, the TRM API may notify the marketplace service 102 of the maximum number of copies of the product that would be possible and indicate the digital asset(s) that have supply shortage for this request.

[00263] Step 2.17-2.18: Start Processing

[00264] The TRM service 402 may generate one or more of the following:

[00265] • Product ID: uniform UUID for products in this set

[00266] • Product Serial IDs: Unique UUID for each individual product copy

[00267] • Serial Numbers: Sequential numbers assigned to each product serial ID (*e.g.*, 0, 1, 2, … (total copies-1))

[00268] A task entry may be created by the TRM service 402 in the task table with status "Processing."

[00269] **Steps 2.19-2.25: Create Assertions, Complete Processing**

[00270] The TRM service 402 may generate one or more of the following assertions for each product serial ID:

[00271] • Product assertion: hash(concatenation of underlying asset assertions) + serial number.

[00272] • Product ownership assertion: product assertion + ORH ID.

[00273] • Market assertion: product assertion + market metadata.

[00274] Once the assertions are inserted into the TIDAL 106, the TRM service 402 may decrement the remaining supply of underlying assets at the asset table in the DB 108 by the total copies of the product. The new product entries may be inserted into the product table in the DB 108 with associated product information. The task entry may be updated with status "done" and with the product ID. In various embodiments, one or more steps of an assertion creation process consistent with various aspects of the disclosed embodiments may be performed as an asynchronous task

[00275] In some embodiments, an indication of the successful uploading of the digital asset may be communicated to the marketplace service 102 and/or the ORH 400 from the TRM 402.

[00276] Download Phase

[00277] **Figures 6A-6D** illustrate a non-limiting example of a download process consistent with certain embodiments disclosed herein.

[00278] Steps 1.1-1.2: Login

[00279] In some embodiments, the consumer 114 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the consumer 114, issuing the consumer an authentication token if successfully authenticated.

[00280] Step 1.3-1.9: Get Verified Marketplace Products

[00281] The marketplace service 102 may use the TIP service 104 DV services to fetch products. In some embodiments, the marketplace service 102 may use the TIP access token to access the DV services of the TIP service 104. A query (*e.g.*, an SQL query) may be used to fetch the information. This query can include any filter, offset, and/or limit parameters. The requested information may be returned from the DB 108 to the marketplace service 104 via the TRM service 402. Once the marketplace service 102 receives the data, the marketplace service 102 may verify the data on the DB 108 using a TRM verification API. For example, the marketplace service may send a list of product IDs to the TRM service 402 for verification. The TRM service 102 may generate market assertions and check the assertions against the TIDAL 106, returning verified product details to the marketplace service 102. The marketplace service 102 may display verified products to consumers 114, and consumers 114 can browse and purchase available digital products from the marketplace service 102. Depending on the business condition for the product, the consumer may have the right to purchase, rent, and/or subscribe to the product.

[00282] Steps 1.10-1.11: Create Transaction

[00283] The marketplace service 102 may receive a request to purchase a product (*e.g.*, for sale, for rent, etc.) from the consumer 114. The marketplace service 102 may make a request to the API of the TRM service 104 to record the transaction, which may include various details relating to the transaction along with a TIP access token of the marketplace service 102.

[00284] Steps 1.12-1.14: Verify Marketplace

[00285] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00286] Steps 1.15-1.19: Create Assertions

[00287] For sellout of public or private products, the TRM service 402 may create a new product ownership assertion, which may comprise the product assertion and the new owner's ID (*e.g.*, the consumer's 114 ID). The TRM service 402 may register the new product ownership assertion to the TIDAL 106. In some embodiments, the prior product ownership assertion at TIDAL 106 may be set to false, reflecting the update in the ownership of the product. For rent or subscription of private products, the TRM service 402 may create a rights assertion, which may comprise the product assertion, an indication of a transaction process, a business condition, and the consumer ID. The rights assertion may be signed by TRM service 402, which may register the rights assertion to the TIDAL 106. Consistent with various embodiments disclosed herein, the TIDAL's ingestion policy may be configured to accept assertions signed by the TRM service 402. The TRM service 402 may use the rights assertion for verification purposes for rent and/or subscription of private digital assets.

[00288] Steps 1.20-1.22: Update DB

[00289] The transaction ID may be generated (*e.g.*, as a UUID), and the data may be recorded in the transaction table in the DB 108. If the product is purchased, the owner ID of the product may also be updated in the product table in the DB 108.

[00290] **Steps 1.24-1.28: Process Payment**

[00291] The marketplace service 102 may redirect the consumer 114 to a third-party secure payment gateway 112 (*e.g.*, PayPal). The consumer 114 may select the payment method, and after successful payment, the marketplace service 102 may distribute the payment to the respective owner(s) (*e.g.*, owner 100) as per any applicable business agreement(s).

[00292] For example, as illustrated in **Figure 6D**, the marketplace service 102 may call an API of the payment gateway service 112. The payment gateway service 112 may respond to the marketplace service 102 with one or more available payment options. The consumer 114 may select a suitable payment method supported by the marketplace service 102 such as, for example and without limitation, credit and/or debit card payment, PayPal, and/or the like. The consumer 114 may confirm payment details with the marketplace service 102.

[00293] The marketplace service 102 may call an API of the payment gateway service 112 with the requested order information and may receive an indication from the payment gateway service 112 indicating successful payment by the consumer 114. The marketplace service 102 may indicate to the consumer 114 the payment was a success. After successful payment, the marketplace service 102 may receive the amount from the consumer 114. The marketplace service 102 may then distribute the payment to respective owner(s) 100 in accordance with one or more applicable business agreements.

[00294] **Asset Playback Within a Product**

[00295] **Figures 7A-7C** illustrate a non-limiting example of an asset playback process consistent with certain embodiments disclosed herein.

[00296] Steps 2.1-2.2: Request Playback

[00297] When a consumer 114 opens a link to access a product, initiating a request to access a digital asset within a product, the marketplace service 102 may send a verification request to the TRM service 402. The request may include, for example and without limitation, a TIP access token (*e.g.*, a marketplace TRM access token), a product serial ID, asset ID(s), and/or a consumer ID.

[00298] Steps 2.3-2.5: Verify Marketplace

[00299] The marketplace TIP access token included in the access request may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00300] Steps 2.6-2.9: Check if User is ORH of Asset

[00301] The product and/or asset data may be retrieved from the product table and asset tables included in the DB 108 respectively. A corresponding asset ownership assertion may be generated by the TRM 402 and checked against the TIDAL 106. For example, if the asset's ORH ID matches the consumer ID, the consumer may be allowed to playback the content, pending asset ownership assertion verification with the TIDAL 106. However, if the remaining supply of the asset is 0, the ORH may not be allowed to playback the asset, as copies of the asset may now have been bundled into products (if the ORH owns any of these products, the ORH may still be able to playback the asset, as these checks may be performed in the subsequent steps).

[00302] Steps 2.10-2.13: Check if User Has Rights for the Product

[00303] Transaction data associated with the consumer 114 and/or product may be retrieved from the DB 108 and provided to the TRM service 402. The TRM service 402

may generate associated rights assertion and verify the rights assertion against the TIDAL 106.

[00304] For example and without limitation, if the product type is 'public', consumers may be allowed to playback the assets within the product. However, if the product type is 'private', the TRM service 402 may check if consumer ID matches an owner ID of the product. If it does, the consumer 114 is allowed to playback the asset, pending product ownership assertion verification in the TIDAL 106.

[00305] The TRM service 402 may further check if the consumer ID is in the invitee's column of the product table. If it is, the consumer 114 may be allowed to playback the asset, pending product invitee assertion verification in a TIDAL 106.

[00306] In addition, the TRM service 402 may check if there is a valid transaction granting the consumer 114 to access the product in the transaction table in DB 108. If there is, the consumer may be allowed to playback the asset, pending rights assertion verification in the TIDAL 106.

[00307] Steps 2.14-2.18: Prepare Asset Playback

[00308] In a DRM application (*e.g.*, for audio content, video content, and/or the like), the TRM service 402 may retrieve the KEK stored in the TIP vault and the KID from the asset table. The TRM service 402 may call DRM service APIs associated with a KMS 404 to issue a token for DRM license by providing the KEK and KID. Once the DRM token is issued by the KMS 404 of the DRM service, the TRM service 402 may distribute the token, and file location URL (*e.g.*, from asset table) to the marketplace services 102. The marketplace service 102 may fetch data from file storage 110 and initiate the DRM playback of the digital asset to the consumer 114 (for example, via a media player application integrating DRM functionality).

[00309] In a non-DRM case, the TRM service 402 may retrieve its secret key from the TIP vault and encrypted content key from the asset table. The TRM service 402 may decrypt the encrypted content key with its secret key. The TRM service 402 may retrieve

the encrypted digital asset from the file storage 110 and decrypt it with the content key. The TRM service 402 may make the decrypted digital asset available to the marketplace service 102, where it may be accessed by the consumer 114.

[00310] Update / Delist Phase

[00311] Asset Update

[00312] **Figures 8A-8C** illustrate a non-limiting example of an asset update process consistent with certain embodiments disclosed herein.

[00313] Steps 1.1-1.2: Login

[00314] In some embodiments, the ORH 400 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the ORH 400, issuing the consumer an authentication token if successfully authenticated.

[00315] Steps 1.3-1.6: Update Asset Request

[00316] The marketplace service 102 may retrieve the list of assets uploaded by the ORH 400 from the asset physical dataset at the TIP service 104 using its marketplace TIP access token. The ORH 400 may select one from the list of assets and may update its metadata. For example, the ORH 400 may update the title and/or preview information included in the asset metadata. In some embodiments, an asset file and/or its associated supply may not be updated (although certain implementations may allow such updates). The marketplace service 102 may send a request to the TRM service 402 with the updated asset metadata, asset ID, and/or user ID, along with its TIP access token.

[00317] Steps 1.7-1.9: Verify Marketplace

[00318] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another

user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00319] Steps 1.10-1.13: Verify Asset Ownership

[00320] The asset data may be retrieved from the DB 108 using, for example, the asset ID and/or the marketplace ID. An asset ownership assertion may be generated with the provided user ID and checked against the TIDAL 106. If the asset ownership assertion is not verified by TIDAL 106, the asset update request may be rejected by the TRM service 402.

[00321] Step 1.14: Verify Asset in Not Bundled into Any Product

[00322] The TRM service 402 may check whether the asset's remaining supply is less than its initial supply. This may be a proxy for checking whether the asset has been bundled into any products or not. If the asset has been bundled into a product that may have been sold already, the ORH 400 may not be allowed to update the asset metadata, as the owner of such a product had purchased the product with the original asset metadata. Thus, if the asset's remaining supply is less than its initial supply, the update operation may be rejected.

[00323] Steps 1.15-1.18: Update Asset

[00324] If a preview file is provided as part of the update, the preview file may be updated in the file storage 110. The asset metadata and preview file URL may be updated in the asset table in DB 108. In some embodiments, an indication of the successful uploading of the digital asset may be communicated to the marketplace service 102 and/or the ORH 400 from the TRM 402.

[00325] Product Update

[00326] In some implementations, certain assets may be limited in their ability to be updated. For example, in certain embodiments, assets bundled in a product may not be

updated, nor may the number of copies available be updated. In some business metadata (*e.g.*, business conditions, sell price, rent price, etc.), product type (*e.g.*, public/private), product metadata (*e.g.*, title, description, etc.), and/or preview file may be updated for all copies of the product owned by the owner 100 performing the update request. **Figures 9A-9C** illustrate a non-limiting example of a product update process consistent with certain embodiments disclosed herein.

[00327] Steps 2.1-2.2: Login

[00328] In some embodiments, the owner 100 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the owner 100, issuing the consumer an authentication token if successfully authenticated.

[00329] Steps 2.3-2.6: Update Product Request

[00330] The marketplace service 102 may retrieve the list of products owned by the owner 100 from the products physical dataset at the TIP service 104 using its marketplace TIP access token. The owner 100 may select one from the list of products and may update it. For example, the owner 100 may update product metadata (*e.g.*, title, description, etc.), preview, asset type, business metadata (*e.g.*, business conditions, sell price, rent price and/or period, subscription price and/or period, rent-ownership price and/or period, and/or the like. In certain embodiments, the assets bundled in a product may not be updated, nor may the number of copies be updated, although other alternatively implementations are also contemplated. The marketplace service 102 may send a request to the TRM service 402 with the updated fields, product ID, owner ID, along with its TIP access token.

[00331] Steps 2.7-2.9: Verify Marketplace

[00332] The marketplace TIP access token included in the product creation request may be verified via an IAM service of the TIP service 104. The marketplace ID may be

obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00333] Step 2.10: Start Processing Update

[00334] In some embodiments, a task entry may be created in the task table with status "processing" and the product ID.

[00335] Step 2.11-2.19: Process Product Update

[00336] In various embodiments, one or more steps of a product update process may be performed as an asynchronous task. The product data corresponding to product ID and owner ID may be retrieved from the product table in the DB 108. For each such product serial ID, a product ownership assertion may be generated and verified with the TIDAL 106. If one or more product ownership assertions are not verified, the product update operation may be rejected. An old market assertion (*e.g.*, product assertion + original market metadata) may be generated and set to false in the TIDAL 106. A new market assertion (*e.g.*, product assertion + new market metadata) may be generated. The new market assertion may be inserted into the TIDAL 106. If the preview file is supplied by the owner 100, the preview may be updated in file storage service 110. The product data and preview may be updated in the DB 108, and the task entry may be set to "done."

[00337] In some embodiments, an indication of the successful uploading of the digital asset may be communicated to the marketplace service 102 and/or the owner 100 from the TRM 402.

[00338] Delist Product

[00339] **Figure 10A-10B** illustrates a non-limiting example of a product delist process consistent with certain embodiments disclosed herein.

[00340] Step 3.1-3.2: Login

[00341] In some embodiments, the owner 100 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the owner 100, issuing the consumer an authentication token if successfully authenticated.

[00342] Steps 3.3-3.5: Delist Product Request

[00343] The marketplace service 102 may retrieve the list of products owned by the owner 100 from the products physical dataset at the TIP service 104 using its marketplace TIP access token. The owner 100 may select one from the list of products. The owner 100 may delist the product if it is currently available on the marketplace service 100 (*e.g.*, set delist value to "1" indicating true), and relist it on the marketplace service 102 if it is not (*e.g.*, set delist value to "0" indicating false). The marketplace service 102 may send a request to the TRM service 402 with the updated delist value, product ID, owner ID, along with its TIP access token.

[00344] Steps 3.6-3.8: Verify Marketplace

[00345] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00346] Step 3.9: Create Task Entry

[00347] A task entry may be created in the task table in the DB 108 with status "processing" and the product ID.

[00348] Steps 3.10-3.12: Verify Product Ownership

[00349] In various embodiments, one or more steps of a product ownership verification process may be performed as an asynchronous task. The TRM service 104

may retrieve entries in the product table in DB 108 corresponding to a specified product ID and/or owner ID. The TRM service 104 may generate the product ownership assertions for the specified owner ID and verify the generated assertions with the TIDAL 106. If any of the product ownership assertions are unverified, the TRM service 402 may reject the delist request.

[00350] Steps 3.13-3.15: Update Products

[00351] The delist property of the corresponding entries in the product table in the DB 108 may be updated to the specified delist value and the task entry may be set to "done". In some embodiments, an indication of the successful delisting may be communicated to the marketplace service 102 and/or the owner 100 from the TRM 402.

[00352] Delete Phase

[00353] Delete Asset

[00354] Figures 11A-11B illustrate a non-limiting example of an asset delete process consistent with certain embodiments disclosed herein. In certain embodiments, an ORH 400 that creates an asset may delete it, as long as it is not bundled into existing products.

[00355] Steps 1.1-1.2: Login

[00356] In some embodiments, the ORH 400 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the ORH 400, issuing the consumer an authentication token if successfully authenticated.

[00357] Steps 1.3-1.5: Delete Asset Request

[00358] The marketplace service 102 may retrieve the list of assets uploaded by the ORH 400 from the asset physical dataset at the TIP service 104 using its marketplace TIP access token. The ORH 400 may select one from the list of assets to delete. The

marketplace service 102 may send a delete request to TRM service 104 with the asset ID, ORH ID, along with its TIP access token.

[00359] Steps 1.6-1.8: Verify Marketplace

[00360] The marketplace TIP access token included in the product creation request may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00361] Steps 1.9-1.11: Verify Asset Ownership

[00362] The asset data may be retrieved from the asset table in the DB 108. The asset ownership assertion may be generated with the provided ORH ID. If the asset ownership assertion is not verified by the TIDAL 106, the asset delete request may be rejected by the TRM service 104 as, in some implementations, only the ORH 400 may delete their assets.

[00363] Step 1.12: Verify Asset is not Bundled into any Product

[00364] The TRM service 402 may check whether the asset's remaining supply is less than its initial supply. This may be a proxy for checking whether the asset has been bundled into any products or not. If the asset has been bundled into a product that may have been sold already, the ORH 400 may not be allowed to delete the asset. Thus, if asset's remaining supply is less than its initial supply, the delete operation may be rejected.

[00365] Steps 1.13-1.17: Delete Asset

[00366] The TRM service 104 may set the asset and asset ownership assertions to false in the TIDAL 106. The encrypted digital asset file and preview file may be deleted from the file storage service 110. Finally, the asset may be deleted from the asset table and

related tasks may be deleted from the task table in the TB 108. In some embodiments, an indication of the successful delisting may be communicated to the marketplace service 102 and/or the owner 100 from the ORH 400.

[00367] Delete Product

[00368] Figures 12A-12B illustrate a non-limiting example of an asset unbundling process consistent with certain embodiments disclosed herein. In certain embodiments, deleting a product may unbundle the assets that are bundled within that product. For example, the supply of each asset bundled in the product to be deleted may be incremented by the total supply of the product. In some implementations, deletion of products may be limited to the ORH 400 that created the products and owns the copies of the product.

[00369] Steps 2.1-2.2: Login

[00370] In some embodiments, the ORH 400 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the ORH 400, issuing the consumer an authentication token if successfully authenticated.

[00371] Steps 2.3-2.5: Delete Product Request

[00372] The marketplace service 102 may retrieve the list of products created by the ORH 400 from the products physical dataset at the TIP service 104 using its marketplace TIP access token. The ORH 400 may select at least one product from the list of products to delete. The ORH 400 may decide to delete one of their products uploaded to the marketplace service 102. The marketplace service 102 may send a request to TRM service 104 with the product ID, ORH ID, along with the marketplace TIP access token.

[00373] Steps 2.6-2.8: Verify Marketplace

[00374] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00375] Step 2.9: Create Task Entry

[00376] The task entry may be created in the task table in the DB 108 with status "processing" and the product ID.

[00377] Steps 2.10-2.14: Verify ORH

[00378] In various embodiments, one or more steps of an ORH verification and/or a product deletion process may be performed as an asynchronous task. The TRM service 402 may retrieve entries in product table corresponding to the specified product ID from the DB 108. If the ORH ID of the product entries does not match the supplied ORH ID, the operation may be rejected. If the owner ID of any product entry does not match the supplied ORH ID, the operation may be rejected. In some embodiments, the ORH 400 may need to own all copies of the product to be deleted, although other implementations where this is not the case are also contemplated.

[00379] Product ownership assertions may be generated by the TRM service 402 for each product serial ID and checked against the TIDAL 106. If any of the product ownership assertions are unverified by the TIDAL 106, the delete request may be rejected.

[00380] Steps 2.15-2.19: Delete Product

[00381] The product assertion, product ownership assertion, and market assertion may be generated for each product serial ID and set to false in the TIDAL 106. The remaining supply of each underlying asset in the product may be incremented in the DB 108 by the total copies of the product to be deleted. The asset table may be updated in the DB 108 to

reflect the new remaining supply counts for the underlying assets. The status of the entries in the task table in the DB 108 may be set to "deleted" and the delete value in the product table in the DB 108 for the product entries may be changed to true.

[00382] In some embodiments, an indication of the successful deletion may be communicated to the marketplace service 102 and/or the owner 100 from the ORH 400.

[00383] Invite/Disinvite Friend for Product

[00384] Invite Process

[00385] **Figures 13A-13C** illustrates a non-limiting example of a product invite process consistent with certain embodiments disclosed herein.

[00386] Steps 1.1-1.2: Login

[00387] In some embodiments, an owner 100 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the owner 100, issuing the owner 100 an authentication token if successfully authenticated.

[00388] Steps 1.3-1.7: Invite Friend to Product

[00389] The marketplace service may retrieve a list of products owned by the owner 100 from the products physical dataset at TIP service 104 using its marketplace TIP access token. The owner 100 may select one from the list of products to invite another user to use, who may be generally referred to herein as a "friend." The owner 100 may add the e-mail address of the friend to invite them to access the product. The marketplace service 102 may check whether the e-mail address exists in the marketplace user table and may retrieve the associated user ID. If it doesn't, the marketplace service 102 may reject the request. The marketplace service 102 may send a request to the TRM service 402 with the user ID , product ID, owner ID, along with its TIP access token.

[00390] Steps 1.8-1.10: Verify Marketplace

[00391] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00392] Steps 1.11-1.17: Process Product Invite request

[00393] The product data corresponding to the product ID and/or owner ID may be retrieved from the product table in the DB 108. For each such product serial ID, a product ownership assertion may be generated and verified with TIDAL 106. If one or more product ownership assertions are not verified, the product invite operation may be rejected. A product invite assertion (*e.g.*, product ownership assertion + invitee ID) may be generated and inserted into the TIDAL 106. The invitee ID may be added to the invitees column in the product table in DB 108. A success message may be communicated from the TRM service 402 to the marketplace service 102.

[00394] Steps 1.18-1.19: Generate Link

[00395] The marketplace service 102 may generate a link to access the asset and send it to the owner 100. The owner 100 can share the link to the friend via e-mail or any other suitable communication method. When someone clicks on the link, it may take them to the marketplace service 102 login screen. After the login that person may play and/or view the asset.

[00396] Disinvite Process

[00397] **Figures 14A-14B** illustrates a non-limiting example of a product disinvite process consistent with certain embodiments disclosed herein.

[00398] Steps 2.1-2.2: Login

[00399] In some embodiments, the owner 100 may login to the marketplace service 102. In certain embodiments, a Google account (*e.g.*, Open ID connect) may be used for login, although other methods of login may also be employed. The marketplace service 102 may authenticate the owner 100, issuing the consumer an authentication token if successfully authenticated.

[00400] Steps 2.3-2.6: Disinvite Friend to Product

[00401] The marketplace service 102 may retrieve the list of products owned by the owner 100 from the products physical dataset at TIP service 104 using its marketplace TIP access token. The owner 100 may select one from the list of products and may update it. The owner 100 may remove the e-mail address of the friend to disinvite them from access to the product. The marketplace service 100 may send a request to the TRM service 402 with the updated user ID , product ID, owner ID, along with its TIP access token.

[00402] Steps 2.7-2.9: Verify Marketplace

[00403] The marketplace TIP access token may be verified via an IAM service of the TIP service 104. The marketplace ID may be obtained from IAM services and/or another user information endpoint. The marketplace service 102 may then be verified as being registered in the DB 108 (*e.g.*, the marketplace ID exists in the marketplace table, as the marketplace table stores the list of verified marketplaces).

[00404] Steps 2.10-2.17: Process Product Disinvite Request

[00405] The product data corresponding to the product ID, owner ID may be retrieved from the product table in the DB 108. For each such product serial ID, a product ownership assertion may be generated and verified using the TIDAL 106. If one or more product ownership assertions are not verified, the product disinvite operation may be rejected. A product invite assertion (*e.g.*, product ownership assertion + invitee ID) may be generated, made false, and registered in the TIDAL 106. The invitee ID from the invitee's column in the product table in the DB 108 may be removed.

[00406] Token Rights Management Using a Commercial Blockchain Ledger

[00407] Although various embodiments of the disclosed systems and methods are described in connection with applications implementing TIDALs and/or TIDAL derivatives, certain embodiments of the disclosed systems and methods may be used in connection with established blockchain ledgers and/or derivatives of such ledgers that may be used in connection with a variety of other applications and/or users, entities, and/or parties. For example, certain non-limiting examples described herein may be used to manage tokens using the FLOW blockchain ledger. Although certain examples are described as being used in connection with a FLOW blockchain ledger, it will be appreciated that various aspects of the disclosed systems and methods may be implemented in connection with a wide variety of blockchain ledgers.

[00408] As discussed above, embodiments of the disclosed systems and methods may allow for rights binding and packaging and/or protection operations in connection with certain NFT transactions (*e.g.*, minting NFTs, transferring NFTs, etc.). For example, uploaded assets (*e.g.,* video, audio, etc.) may be encrypted with a unique key (*e.g.*, a KID, which may be an identifier for a unique key used to encrypt and asset). The KID and usage conditions may be bound together and recorded in a blockchain. For example, an asset token may bind the KID with the uploader's ID (*e.g.,* ORH ID), a product token may bind the KID with the owner's ID (*e.g.*, owner ID) and other usage conditions (*e.g.,* expiry date, invitee ID), and a rent token may bind the KID with the renter ID (*e.g.*, renter ID) and other usage conditions (*e.g.*, expiry date). In some implementations, an asset token may be used to bind the KID with the ORH ID and/or owner ID. Ownership of an NFT may be transferred by checking expiry data in a product token (and/or an asset token depending on the implementation).

[00409] Various embodiments disclosed herein may support an ownership-rent period. An ownership-rent period may be set by an ORH to limit how long a product may be made available for sales, rental, and/or playback in the marketplace. After this period, the product may not be available in the marketplace. In some embodiments, an

ownership-rent period may be associated with a product token enabling sales, rental, access invites, and/or playback during the specified period.

[00410] Further embodiments disclosed herein may provide for a rights binding service. A rights binding service may use TIDALs and/or other blockchains to bind additional rights to an NFT. These rights can include DRM rights and associated metadata can specify DRM authorities. Further embodiments provide for packaging and protection services. In conventional NFT platforms, the underlying content associated with an NFT is typically not part of the NFT. A packaging and protection service can take the original content and transform it, consistent with the metadata in the NFT, to provide various protections (*e.g.*, encryption, authentication, and/or key management for keys that provide security).

[00411] In a FLOW blockchain ledger, objects such as fungible tokens ("FTs"), NFTs, and/or collections may be implemented as resources, which may be stored within a user's account(s). In embodiments where a FLOW ledger is used to implement various aspects of the disclosed token rights management systems and methods, resources may comprise, for example and without limitation, one or more of an:

[00412] • Asset token – An asset token may represent an NFT for an asset. An asset token may comprise a number of fields including, for example and without limitation, one of more of: (1) an asset token ID, which may be immutable and may comprise a unique identifier; (2) a serial number, which may be immutable and may comprise a unique serial number for different copies of an asset; (3) an asset type, which may be mutable and may be either "private" or "public"; (4) a URL, which may be immutable and may comprise a URL or URLs related to an asset file (*e.g.*, an encrypted asset file); and (5) a KID, which may be immutable and may comprise a KID for a DRM management system (*e.g.*, ExpressPlay).

[00413] • Rent token – A Rent token may represent an NFT for renting an asset, enabling its owner to playback the content of the rented asset before an expiry timestamp. A rent token may comprise a number of fields including, for example and without

limitation, one or more of: (1) a rent token ID, which may be immutable and may comprise a unique identifier; (2) an asset token ID, which may be immutable and may comprise the asset token ID of the asset being rented; (3) a URL, which may be immutable and may comprise a URL or URLs related to an asset file (*e.g.*, an encrypted asset file); (4) a KID, which may be immutable and may comprise a KID for a DRM management system (*e.g.,* ExpressPlay); and (5) an expiry timestamp, which may be immutable and comprise a timestamp after which the rent token becomes invalid for playing back the content. The expiry timestamp may be initialized to the current data plus the rental period.

[00414] • Asset collection – An asset collection may comprise a collection of owned asset tokens.

[00415] • Rent collection – A rent collection may comprise a collection of owned rent tokens.

[00416] • Sale collection – A sale collection may comprise a resource used to track the asset tokens that are available for sale and rent. The sale collection may track the asset prices, enabling the owner to list, delist, and/or update listings. Purchase and rent transactions may be enabled via the sale collection. A cut of each sale/rent transaction may be sent to an administrator's cryptographic currency vault and/or digital wallet (*e.g.*, using a fiat-backed stablecoin issued as a fungible token on the FLOW network such as FUSD). In some embodiments, a specific cut percentage (*e.g.,* 5%) may be updated by the administrator at any time. Alternatively and/or in combination, logic may accommodate separate cuts for the marketplace and the TRM administrator. It will be appreciated that a variety of other cut percentages may be provided to a variety of parties and/or multiple parties in further embodiments.

[00417] • The content of URL in asset token and rent token may depend on the implementation. For example and without limitation, it may comprise: (1) a link to the asset in a file system protocol (*e.g.*, Interplanetary File System ("IPFS"); (2) a link to an

JSON file which comprises copies of the asset in IPFS; and/or (3) one or more links to a manifest file (*e.g.,* DASH, HLS, etc.).

[00418] Asset Creation

[00419] **Figure 15A-15F** illustrate a non-limiting example of an asset creation process using a blockchain ledger consistent with certain embodiments disclosed herein. In various embodiments, the FLOW blockchain ledger may be used, although it will be appreciated that embodiments of the disclosed systems and methods may be used in connection with a variety of other blockchain ledgers.

[00420] In various embodiments, a wallet service 1500 (e.g., a cryptocurrency wallet) and a blockchain ledger 1502 may be used in connection with various embodiments of the disclosed systems and methods and be conceptually included in blockchain functionality of the disclosed ecosystem. Package functionality in the disclosed ecosystem may comprising, in some embodiments, a storage service 1502, a media packager 1506, and a distributed event store and/or stream processing platform such as Kafka 1508. TRM system functionality may be implemented using an orchestrator service 110, a TIP IAM and/or data service ("DS") 1512, and a blockchain connector service 1514. A secure DB 1516 may also be used to store and/or manage various tables and/or other values. Various interactions between these systems and/or services are described in more detail below.

[00421] Steps 1.1 – 1.3: User Login

[00422] The ORH 400 may login to the wallet service 1500 via the marketplace service 102. In the FLOW blockchain, the marketplace service 102 may use the FLOW Client Library ("FCL") to authenticate the ORH 400 with the wallet service 1500. FCL may comprise a JavaScript Client Library, enabling applications to integrate with FCL-compatible wallets (*e.g.,* Blocto, Ledger, Dapper Wallet, etc.). The marketplace service 102 may obtain details about the ORH account like a wallet ID after the authentication. The marketplace service may check whether the wallet ID matches the registered ORH wallet IDs at the marketplace service 102.

[00423] Steps 1.4 – 1.5: Upload Asset to Marketplace

[00424] The ORH 400 may upload the asset content (*e.g.,* image, video, text, audio, etc.) at the marketplace service 102 and may specify one or more of the following for the asset: (1) number of copies of the asset to mint; (2) a title of the asset; (3) a preview file (*e.g.,* a thumbnail); (4) an asset type, which may comprise a public asset where any consumer can access (*e.g.,* view and/or listen) the asset, or a private asset where a consumer can access (*e.g.,* watch or listen) the asset after purchasing transactions (*e.g.,* buyout, rent, etc.); and (5) a description and/or any other metadata associated with an asset that may be defined by the marketplace service. The ORH 400 may further upload a preview file to serve as a thumbnail for the asset on the marketplace service 102 portal.

[00425] Steps 1.6-1.8: Upload Asset Media Content to Package Function

[00426] The marketplace service 102 may upload the asset (*e.g.*, image, video, music) content to a storage service 1504 (*e.g.*, AWS S3) and may receive a URL and transaction ID. For certain content (*e.g.*, audio and/or video), the marketplace service 102 may also subscribe to the Kafka service 1508 which may notify a media packager service 1506 when the digital asset content is uploaded and if there is any update to the digital asset content.

[00427] Steps 1.9-1.11: Upload Asset to TRM

[00428] The marketplace service 102 may obtain an access token from the TIP IAM service 1512 service using its API key and secret. The marketplace service 102 may provide an orchestrator service 1510 an asset upload request that may comprise one or more of: (1) an ORH wallet ID; (2) information set by the ORH 400 described above (*e.g.*, number of copies of the asset, title of the asset, preview file, asset type, etc.); and (3) an access token; (4) the transaction ID.

[00429] Steps 1.12-1.16: Verify Marketplace and Create Task Entry

[00430] The orchestrator service 1510 may verify the marketplace access token with the TIP IAM service 1512. The marketplace group information may be provided to the orchestrator service 1510 from the TIP IAM service 1512. The orchestrator service 1510 may verify whether the marketplace service 102 is a member of the marketplace group. A task entry may be created by the orchestrator service 1510 in the task table with the status "processing" in the secure DB 1516 while the asset encryption process takes place within the package function services 1505-1508. The orchestrator service 1510 may then send an acknowledgement message to the marketplace service 102 and/or the ORH 400.

[00431] Steps 1.17-1.21 (Asynchronous Task): Package Asset Content, Generate KID, and Fact

[00432] For certain content (*e.g.*, video and/or audio content), the Kafka services 1508 may inform the media packager services 1506 when the asset has finished uploading. The media packager services 1506 may encrypt the content file and save the encrypted asset content on the file storage service 1504 (*e.g.* S3), then the media packager service 1506 may return KID, URL(*), and the transaction ID to the orchestrator service 1510. In some embodiments, the orchestrator service 1510 may use the transaction ID to correlate KID and URLs and/or ORH ID and/or other information set by the ORH 400.

[00433] To protect the content, a secret key may be obtained from TIP services vault. A new secret key may be generated and stored in the TIP services vault if it does not already exist. A content key may be generated and the asset file may be encrypted using the content key. The encrypted asset may be stored by the file storage service 1504. The content key may be encrypted using the secret key. An entry may be created at the asset table including the asset information at the DB 1516. The entry may be updated at the task table with the status set to "done." The asset ID may also be updated in the task table at the DB 1516.

[00434] The orchestrator service 1510 may generate an asset fact. The asset fact may comprise one or more of an owner ID (*e.g.*, an ORH Wallet ID), an asset type, a KID, a URL, and/or a serial number. The content of the URL may depend on the

implementation and may comprise, for example and without limitation, one or more of a link of the asset in IPFS, a link to JSON which may comprise copies of the asset in IPFS, one or more links to a manifest file (*e.g.,* DASH, HLS), and/or the like.

[00435] Step 1.22-1.25 (Asynchronous Task): Initiate Asset Token Creation

[00436] The orchestrator service 1510 may provide the asset fact to a blockchain connector service 1514 to initiate asset token creation by the blockchain 1512, which may comprise the FLOW blockchain. The blockchain connector service 1514 may retrieve the blockchain credentials (*e.g.,* a private key) from the TIP service vault. A transaction to request minting the asset token may be generated by the blockchain connector service. The transaction may be signed with the private key as proposer and payer. The blockchain connector service may send the signed transaction to the blockchain 1514 (*e.g.*, via the orchestrator service 1510). For example, in some embodiments, the blockchain connector service 1510 may call a mintNFT function in a smart contract at the FLOW blockchain by providing the signed transaction.

[00437] Step 1.26 (Asynchronous Task): Mint Asset Token

[00438] Provided with the signed transaction, the smart contract may transfer the newly minted asset token into ORH's blockchain account storage. Remaining objects may be saved in the asset fact as metadata. The minted asset token may be configured to make the asset token ID, KID, URL, and/or serial number to be immutable (*e.g.,* read only) so that certain values may be securely bound to the asset token with its owner identified with the owner wallet ID and/or ORH wallet ID. These bound values may comprise, for example and without limitation, one or more of an asset token ID, which may be immutable, and metadata, which may comprise one or more of an asset type, a KID, which may be immutable, a URL, which may be immutable, and/or a serial number, which may be immutable.

[00439] Step 1.27-1.29 (Asynchronous Task): Create Asset Entry

[00440] The blockchain connector service 1514 may receive the update asset token minting result by polling the blockchain 1512 and make sure the update with the orchestrator service 1510. Once the blockchain connector service 1514 receives the result, it may create a new entry in the asset table in the DB 1516 with all the newly created asset information. The task entry in the task table may be changed in the DB 1516 to the status "done." After successful update, the marketplace service 102 can fetch the asset details from the DB 1516. In some implementations, after minting the asset token, this asset may not be listed on the marketplace service 102 until the ORH 400 updates the asset to list the asset for the marketplace service 102.

[00441] Step 1.30: Delete the Original Asset Content File

[00442] The marketplace service 102 may delete the originally uploaded asset content file at the file storage (*e.g.*, AWS S3 storage).

[00443] Update Asset

[00444] **Figures 16A-16C** illustrate a non-limiting example of an asset update process using a blockchain ledger consistent with certain embodiments disclosed herein.

[00445] Steps 1.1-1.2: User Login

[00446] An owner 100 may log into a wallet service 1500 via a marketplace service 102. In the FLOW blockchain, the marketplace service 102 may use FCL to authenticate an owner 100 with the wallet service 1500. The marketplace service 102 may obtain details about the owner account corresponding to the owner 100 like a wallet ID after the authentication

[00447] Steps 1.3-1.6: Retrieve Asset Information

[00448] The marketplace service 102 may obtain an access token from the TIP IAM service 1512 using its API key and secret. The marketplace service 102 may fetch the list of assets created by the owner 100 from DB 1516 via the TIP DS service 1512 using its access token.

[00449] Steps 1.7-1.8: Request Asset Update

[00450] The owner 100 may select an asset from the list and can choose to update its asset information by sending an associated request to the marketplace service 102. For example and without limitation, an owner 100 may update one or more of the following:

[00451] • Title

[00452] • Asset type – An asset type may be public, where any consumers can access (*e.g.,* watch and/or listen) the asset, or private, where consumers can access (*e.g.,* watch and/or listen) the asset after completing purchase transactions (*e.g.,* buyout, rent, etc.).

[00453] • Business condition – Business conditions may comprise, for example and without limitation, one or more of sellout and/or rent. If sellout and/or rent is selected as a business condition, the asset may be listed to the marketplace service 102 after updating the asset. If neither of sellout and/or rent are selected as a business condition, the asset may be delisted from the marketplace service after updating the asset.

[00454] • Rental period – This may be set when the business condition includes "Rent".

[00455] • Price – Price may include, for example and without limitation, a price for sellout, a price for rent, etc.

[00456] • Description, and/or any other metadata defined by the marketplace service 102.

[00457] The marketplace service 102 may send the updated information along with the owner wallet ID, asset token, ID, and/or access token to the orchestrator service 1510

[00458] Steps 1.9-1.11: Verify Marketplace

[00459] The orchestrator service 1510 may verify the marketplace access token using the TIP IAM service 1512. Marketplace group information may be obtained by the

orchestrator service 1510 from the TIP IAM service 1512 The orchestrator service 1510 may verify whether the marketplace is a member of the marketplace group.

[00460] Step 1.12: Generate Update Fact

[00461] The orchestrator service 1510 may generate the update fact. The update fact may be an object that comprises one or more of an owner wallet ID, an access token ID, and/or update data.

[00462] Steps 1.13-1.15: Initiate Asset Update

[00463] The orchestrator service 1510 may provide the update fact to the blockchain connector service 1514 to initiate the asset update at the blockchain 1502. The blockchain connector service 1514 may generate a ttansaction to request updating the asset token and/or sale collection. The blockchain connector service 1514 may further send the transaction to the marketplace service 102 via the orchestrator service 1510.

[00464] Steps 1.16-1.20: Update Asset at Blockchain

[00465] The marketplace may request the owner 100 to sign the transaction via the wallet service 1500. Once the owner 100 confirms the transaction, the wallet service 1500 may sign the transaction with the owner's signing key. The wallet service may send the signed transaction to the blockchain 1502. Given the signed transaction, the smart contract may update the asset token and/or sale collection depending on the update request.

[00466] Steps 1.21-1.23: Update Asset Entry

[00467] The blockchain connector service 1514 may retrieve the asset update result by polling the blockchain 1502. Once the blockchain connector service 1514 gets the result, it may call the orchestrator service API and may provide transaction details to the orchestrator service 1510. The orchestrator service 1510 may save the updated result in DB 1516. The blockchain 1502 may send a success message to the wallet service 1500 which then may send it to the marketplace service 102 and/or owner 100.

[00468] Sellout Asset

[00469] **Figures 17A-17B** illustrate a non-limiting example of an asset purchase process using a blockchain ledger consistent with certain embodiments disclosed herein.

[00470] Steps 1.1-1.4: Retrieve Asset Information

[00471] The marketplace service 102 may obtain an access token from the TIP IAM service 1512 using its API key and secret. A list of listed assets may be fetched by the marketplace service 102 from the dataset by TIP DS service 1512 using its access token.

[00472] Steps 1.5-1.6: User Login

[00473] The buyer 1700 may login to the wallet service 1500 via the marketplace service 102. In the FLOW blockchain, the marketplace service 102 may use FCL to authenticate the buyer 1700 with the wallet service 1700. The marketplace service may obtain details about the account associated with the buyer 1700 like a wallet ID after the authentication.

[00474] Steps 1.7-1.8: Request Asset Purchase

[00475] The buyer 1700 may select an asset from the list and can choose to purchase the asset. The marketplace service 102 may send the owner wallet ID for the asset, asset token ID, and/or access token to the orchestrator service 1510.

[00476] Steps 1.9-1.11: Verify Marketplace

[00477] The orchestrator service 1510 may verify the marketplace access Token at the TIP IAM service 1512. Marketplace group information may be obtained by the orchestrator service 1510 from the TIP IAM service 1512. The orchestrator service 1510 may verify whether the marketplace service 102 is a member of the marketplace group.

[00478] Step 1.12: Generate Purchase Fact

[00479] The orchestrator service 1510 may generate a purchase fact. The purchase fact may comprise one or more of the owner Wallet ID and/or the asset token ID.

[00480] Steps 1.13-1.15: Initiate Asset Transfer

[00481] The Orchestrator may provide a purchase fact to the blockchain connector service 1514 to initiate an asset transfer at the blockchain 1502. The blockchain connector service 1514 may generate a transaction to request transfer of an asset token. The transaction may be sent by the blockchain connector service to the marketplace service 102 via the orchestrator service 1510.

[00482] Steps 1.16-1.21: Purchase Asset at Blockchain

[00483] The marketplace service 102 may request the buyer 1700 to sign the transaction via the wallet service 1500. Once the buyer confirms the transaction, the wallet service 1500 may sign the transaction with the buyer's signing key. The wallet service 1500 may send the signed transaction to the blockchain 1502. Given the signed transaction, the smart contract may transfer ownership of the asset token to the buyer 1700, update the owner's sale collection by removing asset token ID, and initiate a payout (*e.g.*, a payout in FUSD or some other cryptographic currency). In some embodiments, the administrator may receive a percentage of the transaction defined by a cut percentage (*e.g.,* 5% of the sale) and the owner may receive the remaining payment, although it will be appreciated that a variety of other cut percentages may be provided to a variety of parties in further embodiments.

[00484] Step 1.22-1.24: Update Asset Entry

[00485] The blockchain connector service 1514 may determine the asset transfer result by polling the blockchain 1502. Once the blockchain connector obtains the result, it may call the API of the orchestrator service 1510 and provides transaction details to the orchestrator service 1510. The orchestrator service 1510 may save the updated result in the DB 1516. The blockchain 1502 may send a success message to the wallet service 1500 which then may send it to the marketplace service 102 and/or the buyer 1700.

[00486] Rent Asset

[00487] **Figures 18A-18C** illustrate a non-limiting example of an asset rental process using a blockchain ledger consistent with certain embodiments disclosed herein.

[00488] Steps 1.1-1-4: Retrieve Asset Information

[00489] The marketplace service 102 may obtain an access token from the TIP IAM service 1512 using its API key and secret. The marketplace service 102 may fetch the listed assets from the dataset at TIP DS 1512 using its access token.

[00490] Steps 1.1-1.2: User Login

[00491] A renter 1800 may log into a wallet service 1500 via a marketplace service 102. In the FLOW blockchain, the marketplace service 102 may use FCL to authenticate the renter 1800 with the wallet service 1500. The marketplace service 102 may obtain details about an account associated with the renter 1800 such as a wallet ID after the authentication.

[00492] Steps 1.7-1.8: Request Asset Rent

[00493] The renter 1800 may select an asset from a list and can choose to rent the asset, issuing a request to the marketplace service 102. The marketplace service 102 may send the owner wallet ID for the asset, asset token ID, and access token to the orchestrator service 1510.

[00494] Steps 1.9-1.11: Verify Marketplace

[00495] The orchestrator service 1510 may verify the marketplace access token using the TIP IAM service 1512. Marketplace group information may be obtained by the orchestrator service 1510 from the TIP IAM service 1512. The orchestrator service 1510 may verify whether the marketplace service 102 is a member of the marketplace group.

[00496] Step 1.12: Generate Rent Fact

[00497] The orchestrator service 102 may generate a rent fact. The rent fact may comprise one or more of the owner wallet ID and/or the asset token ID.

[00498] **Steps 1.13-1.15: Initiate Asset Rent**

[00499] The orchestrator service 1510 may provide the generated rent fact to the blockchain connector service to initiate minting a rent token at the blockchain 1502. The blockchain connector service 1510 may generate a transaction and send the transaction to the marketplace service 102 via an orchestrator service 1510.

[00500] **Steps 1.16-1.21: Rent Asset at Blockchain**

[00501] The marketplace service 102 may request the renter 1800 to sign the transaction via the wallet service 1500. Once the renter confirms the transaction, the wallet service 1500 may sign the transaction with the renter's signing key. The wallet service may send the signed transaction to the blockchain 1502.

[00502] Provided with the signed transaction, the smart contract may mint a rent token and transfer it to the renter's blockchain account storage. The smart contract may further initiate a payout in (*e.g.*, via a cryptocurrency like FUSD). In some embodiments, the administrator may receive a percentage of the transaction defined by a cut percentage (*e.g.*, 5% of the sale) and the owner may receive the remaining payment, although it will be appreciated that a variety of other cut percentages may be provided to a variety of parties in further embodiments.

[00503] The minted rent token may be configured to make the rent token ID, asset token ID, KID, URL, and/or expiry date be immutable (*e.g.*, read only) so that certain values may be securely bound to the rent token owned by the renter.

[00504] These bound values may comprise, for example and without limitation, one or more of an rent token ID, which may be immutable, and metadata, which may comprise one or more of an asset token ID, which may be immutable, a KID, which may be immutable and/or be the same KID included in the asset token identified with the asset

token ID, a URL, which may be immutable and/or be the same URL included in the asset token identified with the asset token ID, and/or a expiry date, which may be immutable and may be the date after which the rent token becomes invalid for playing back the asset content. In some embodiments, the expiry date may be initialized to the current date plus the rental period specified in the asset token.

[00505] Step 1.22-1.24: Create Transaction Entry

[00506] The blockchain connector service 1514 may get the rent token minting result by polling the blockchain 1502 and share it with the orchestrator 1510. The orchestrator service 1510 may save the updated result in the DB 1516. The blockchain 1502 may send a success message to the wallet service 1500 which then may send it to the marketplace service 102 and/or the renter 1800.

[00507] Playback Asset

[00508] **Figures 19A-19C** illustrate a non-limiting example of an asset playback process using a blockchain ledger consistent with certain embodiments disclosed herein.

[00509] Steps 1.1-1.4: Retrieve Asset Information

[00510] The marketplace service 102 may obtain an access token from the TIP IAM service 1512 using its API key and secret. A list of assets may be fetched by the marketplace service 102 from the DB 1516 using TIP DS services 1512 and its access token.

[00511] Steps 1.5-1.6: User Login

[00512] A consumer 1900 may login to the wallet service 1500 via the marketplace service 102. In the FLOW blockchain, the marketplace service 102 may use FCL to authenticate the consumer 1900 with the wallet service 1500. The marketplace service 102 may obtain details about the consumer account 1900 such as a wallet ID after the authentication.

[00513] Steps 1.7-1.8: Request Asset Playback

[00514] The consumer 1900 may select an asset from the list and can choose to playback the asset by requesting the marketplace service 102. The marketplace service 102 may send the consumer wallet ID for the asset, asset token ID, and/or access token to the orchestrator service 1510.

[00515] Steps 1.9-1.11: Verify Marketplace

[00516] The orchestrator service 1510 may verify the marketplace access token at the TIP IAM service 1512. Marketplace group information may be obtained by the orchestrator service 1510 from the TIP IAM service 1512. The orchestrator service 1510 may verify whether the marketplace service 102 is a member of the marketplace group.

[00517] Step 1.12: Initiate Usage Evaluation

[00518] The orchestrator service 1510 may provide an asset token ID and consumer wallet ID to the blockchain connector service 1514 to initiate running a script.

[00519] Steps 1.14-1.15: Evaluate Usage

[00520] After initiating, the script may evaluate usage of the asset at the blockchain 1502. If the asset type is "public," the process may proceed with a "success" result. If the asset type is "private" and the consumer 1900 is the owner of the asset token, the process may proceed to with a "success" result.

[00521] If the asset type is "private" and the consumer 1900 is not the owner of the asset token, a check may be performed to determine if there is a rent token associated with the asset token and the consumer 1900. If there is no rent token associated with the asset token and consumer 1900, the process may proceed with a "fail" result.

[00522] If there is a rent token associated with the asset token and consumer 1900, the expiry date may be checked in the rent token. If the expiry date is already passed at the

current time, the process may proceed with a "fail" result. If the expiry date is not passed at the current time, the process may proceed with a "success" result.

[00523] Steps 1.16-1.17: Send Evaluation Result

[00524] If there is a "success" result, the KID and URL for the asset token may be sent to the blockchain connector service 1514 from the blockchain 1502, and the blockchain connector service 1514 may forward them to the orchestrator service 1510.

[00525] Steps 1.18-1.24: Playback Asset

[00526] If the orchestrator service 1510 receives a "success" result from the blockchain connector service 1514 along with a KID and/or URL, the orchestrator service 1510 may proceed to engage in transactions that, in some embodiments, may depend on a DRM implementation and/or use case. For example, in at least one DRM use case, which may be used in implementations involving audio, image, and/or video content, the orchestrator service 1510 may retrieve a KEK stored in a TIP vault. The orchestrator service 1510 may call a KMS 404 via DRM service APIs to issue a token for a DRM license by providing the KEK and the KID. The KMS 404may verify the K from the KEK and KID before issuing a DRM token. Once the token is issued by the KMS 404, the orchestrator service 1510 may return to the token and the URL to the marketplace service 102. The marketplace service 102 may use the DRM client (*e.g.*, KMS 404) to redeem the token from the DRM service, fetch the encrypted asset data by using the given URL, and initiate the playback of the asset with the player application.

[00527] In a non-DRM case, the orchestrator service 1510 may retrieve its secret key from the TIP Vault and encrypted content key from the asset table in the DB 1516. The orchestrator service 1510 may decrypt the encrypted content key with its secret key. The orchestrator service 1510 may retrieve the encrypted asset from the given URL and may decrypt it with the content key. The orchestrator service 1510 may then make the decrypted asset available to the marketplace service 102, and the consumer 1900 can access the asset on the marketplace service 102.

[00528] **Figure 20** illustrates a flow chart of a non-limiting example of a method 2000 of managing a digital asset consistent with certain embodiments of the present disclosure. The illustrated method 2000 may be implemented in a variety of ways, including using software, firmware, hardware, and/or any combination thereof. In certain embodiments, various aspects of the method 2000 and/or its constituent steps may be performed by a service, system, and/or device configured to implement embodiments of a TRM service (and/or associated services such as a marketplace service, a TIP service, a TIDAL service, a DRM and/or KMS service, etc.), and/or any other suitable application, device, system and/or service or combination of applications, devices, systems, and/or services.

[00529] At 2002, a TRM service may receive from a marketplace service a DRM data access request. In some embodiments, the TRM service may comprise a service of a TIP service. The DRM access request may comprise, among other things, a marketplace access token and/or an identifier associated with a user. In further embodiments, the DRM access request may further comprise an identifier associated with the digital asset and/or an identifier associated with a product that includes the digital asset. In some embodiments, the DRM access request may be issued by the marketplace service to the TRM service in response to a receiving a user request to access a digital asset from the user associated with the identifier.

[00530] Asset information may be obtained at 2004 that is associated with the marketplace service based on the marketplace access token included in the DRM access request. In some embodiments, the asset information may comprise, among other things, a hash of the digital asset.

[00531] In some embodiments obtaining the asset information associated with the marketplace service may comprise obtaining the asset information from a DB associated with a TIP service. For example, in some embodiments, the marketplace service may be verified with the TIP service using the marketplace access token. Based on successful verification of the marketplace service, an indication may be received from the TIP service on an account identifier associated with the marketplace service, which may be used to obtain the asset information associated with the marketplace.

[00532] At 2006, an asset rights assertion may be generated based on the hash of the digital asset and the identifier associated with the user received in the DRM access request. The TRM service may query the asset rights assertion against a trusted ledger, which may comprise a TIDAL and/or a blockchain ledger at 2008. If the asset rights assertion is recorded in the trusted ledger, it may be determined that the user has rights to access the digital asset. In some embodiments, it may be determined that the user is an owner of the digital asset. Checks may be performed to determine whether there are available remaining copies of the digital asset available for consumption and/or further management.

[00533] Based on the determination, a query may be issued to a DRM service at 2010. In some embodiments, the DRM service may comprise a KMS service (and may in some implementations be a service of the TIP service). In various embodiments, the TRM service, the marketplace service, the TIP service, the DRM service (and/or KMS service), and the TIDAL may execute on a single system and/or any suitable combination of systems.

[00534] The query to the DRM service may comprise a KID and/or a KEK, which may comprise a public encryption key associated with the user. In some embodiments, the KID and/or the KEK may be accessed from a DB associated with the TIP service.

[00535] At 2012, the DRM service may issue a DRM token in response to the query. The DRM token may comprise an encrypted version of a content key that may be used to decrypt the digital asset. The TRM service may send the DRM token and location of the digital asset (*e.g.*, a URL) to the marketplace service at 2014 for communication to a system associated with the user.

[00536] System and/or Service Architecture

[00537] **Figure 21** illustrates a non-limiting example of a system 2100 that may be used to implement certain embodiments of the systems and methods of the present disclosure. The system 2100 of **Figure 21** and/or aspects thereof may be included in a system, service, and/or device associated with an owner, an ORH, a consumer, a buyer, a

renter, a marketplace service, a TRM service, a TIP service, a TIDAL and/or an associated service, a DRM and/or KMS service, a DB, a file storage service, a payment gateway service, a wallet service, a blockchain service, a media packager service, an orchestrator service, a blockchain connector service, and/or any other service, which may comprise a trusted service, system, and/or component configured to implement embodiments of the disclosed systems and methods and/or aspects thereof.

[00538] The various systems and/or devices used in connection with aspects the disclosed embodiments may be communicatively coupled using a variety of networks and/or network connections (*e.g.*, network 2112). In certain embodiments, the network 2112 may comprise a variety of network communication devices and/or channels and may utilize any suitable communications protocols and/or standards facilitating communication between the systems and/or devices. The network 2112 may comprise the Internet, a local area network, a virtual private network, and/or any other communication network utilizing one or more electronic communication technologies and/or standards (*e.g.*, Ethernet or the like). In some embodiments, the network 2112 may comprise a wireless carrier system such as a personal communications system ("PCS"), and/or any other suitable communication system incorporating any suitable communication standards and/or protocols. In further embodiments, the network 2112 may comprise an analog mobile communications network and/or a digital mobile communications network utilizing, for example, code division multiple access ("CDMA"), Global System for Mobile Communications or Groupe Special Mobile ("GSM"), frequency division multiple access ("FDMA"), and/or time divisional multiple access ("TDMA") standards. In certain embodiments, the network 2112 may incorporate one or more satellite communication links. In yet further embodiments, the network 2112 may utilize IEEE's 802.11 standards, Bluetooth®, ultra-wide band ("UWB"), Zigbee®, and or any other suitable standard or standards.

[00539] The various systems and/or devices used in connection with aspects of the disclosed embodiments may comprise a variety of computing devices and/or systems, including any computing system or systems suitable to implement the systems and

methods disclosed herein. For example, the connected devices and/or systems may comprise a variety of computing devices and systems, including laptop computer systems, desktop computer systems, server computer systems, distributed computer systems, smartphones, tablet computers, and/or the like.

[00540] In certain embodiments, the systems and/or devices may comprise at least one processor system configured to execute instructions stored on an associated non-transitory computer-readable storage medium. As discussed in more detail below, systems used in connection with implementing various aspects of the disclosed embodiments may further comprise a secure processing unit ("SPU") configured to perform sensitive operations such as trusted credential and/or key management, cryptographic operations, secure policy management, and/or other aspects of the systems and methods disclosed herein. The systems and/or devices may further comprise software and/or hardware configured to enable electronic communication of information between the devices and/or systems via a network using any suitable communication technology and/or standard.

[00541] As illustrated in **Figure 21**, the example system 2100 may comprise: a processing unit 2102; system memory 2104, which may include high speed random access memory ("RAM"), non-volatile memory ("ROM"), and/or one or more bulk non-volatile non-transitory computer-readable storage mediums (*e.g.*, a hard disk, flash memory, etc.) for storing programs and other data for use and execution by the processing unit 2102; a port 2106 for interfacing with removable memory 608 that may include one or more diskettes, optical storage mediums (*e.g.*, flash memory, thumb drives, USB dongles, compact discs, DVDs, etc.) and/or other non-transitory computer-readable storage mediums; a network interface 2110 for communicating with other systems via one or more network connections and/or networks 2112 using one or more communication technologies; a user interface 2114 that may include a display and/or one or more input/output devices such as, for example, a touchscreen, a keyboard, a mouse, a track pad, and the like; and one or more busses 2116 for communicatively coupling the elements of the system.

[00542] In some embodiments, the system 2100 may, alternatively or in addition, include a trusted execution environment and/or an SPU 2118 that is protected from tampering by a user of the system or other entities by utilizing secure physical and/or virtual security techniques. A trusted execution environment and/or a SPU 2118 can help enhance the security of sensitive operations such as personal information management, trusted credential, token, and/or key management, privacy and policy management, and other aspects of the systems and methods disclosed herein. In certain embodiments, the trusted execution environment and/or SPU 2118 may operate in a logically secure processing domain and be configured to protect and operate on secret information, as described herein. In some embodiments, the trusted execution environment and/or a SPU 2118 may include internal memory storing executable instructions or programs configured to enable the SPU 2118 to perform secure operations, as described herein.

[00543] The operation of the system 2100 may be generally controlled by the processing unit 2102 and/or an SPU 2118 operating by executing software instructions and programs stored in the system memory 2104 (and/or other computer-readable media, such as memory 2108, which may be removable). The system memory 2104 may store a variety of executable programs or modules for controlling the operation of the system. For example, the system memory may include an operating system ("OS") 2120 that may manage and coordinate, at least in part, and/or system hardware resources and provide for common services for execution of various applications.

[00544] The system memory 2104 may further include, without limitation, communication software 2122 configured to enable in part communication with and by the system, one or more applications, a cryptographic operation module 2124 configured to perform various aspects of the disclosed embodiments (*e.g.*, cryptographic key and hash generation operations, key management operations, etc.), one or more verification digital asset and/or token management modules 2126 configured to perform various aspects of the methods disclosed herein, and/or any other information, modules, and/or applications configured to implement embodiments of the systems and methods disclosed herein.

[00545] The systems and methods disclosed herein are not inherently related to any particular computer, electronic control unit, or other apparatus and may be implemented by a suitable combination of hardware, software, and/or firmware. Software implementations may include one or more computer programs comprising executable code/instructions that, when executed by a processor, may cause the processor to perform a method defined at least in part by the executable instructions. The computer program can be written in any form of programming language, including compiled or interpreted languages, and can be deployed in any form, including as a standalone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. Further, a computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

[00546] Software embodiments may be implemented as a computer program product that comprises a non-transitory storage medium configured to store computer programs and instructions, that when executed by a processor, are configured to cause the processor to perform a method according to the instructions. In certain embodiments, the non-transitory storage medium may take any form capable of storing processor-readable instructions on a non-transitory storage medium. A non-transitory storage medium may be embodied by a compact disk, digital-video disk, a magnetic disk, flash memory, integrated circuits, or any other non-transitory digital processing apparatus memory device.

[00547] Although the foregoing has been described in some detail for purposes of clarity, it will be apparent that certain changes and modifications may be made without departing from the principles thereof. It should be noted that there are many alternative ways of implementing both the systems and methods described herein. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein but may be modified with the scope and equivalents of the appended claims.

WHAT IS CLAIMED IS:

1. A method for managing a digital asset performed by a token rights management service executing on a system comprising a processor and a non-transitory computer-readable storage medium storing instructions that, when executed by the processor, cause the system to perform the method, the method comprising:

receiving, from a marketplace service, a digital rights management data access request, the digital rights management data access request comprising a marketplace access token and an identifier associated with a user, the digital rights management data access request being issued by the marketplace service in response to receiving a user request to access a digital asset;

obtaining asset information associated with the marketplace service based on the marketplace access token, the asset information comprising a hash of the digital asset;

generating an asset rights assertion based on the hash of the digital asset and the identifier associated with the user;

determining that the user has rights to access the digital asset by verifying that the asset rights assertion is recorded in a trusted ledger;

issuing a query to a digital rights management service based on determining that the user has rights to access the digital asset;

receiving, from the digital rights management service in response to the query, a digital rights management token associated with the digital asset; and

sending the digital rights management token and a location of the digital asset to the marketplace service for communication to a system associated with the user.

2. The method of claim 1, wherein obtaining the asset information associated with the marketplace service comprises obtaining the asset information from a database associated with a trusted data management service.

3. The method of claim 2, wherein the method further comprises:

verifying the marketplace service with the trusted data management service using the marketplace access token; and

receiving, based on successful verification of the marketplace service, an indication from the trusted data management service of an account identifier associated with the marketplace service,

wherein obtaining the asset information associated with the marketplace service is based on the account identifier associated with the marketplace service.

4. The method of claim 1, wherein the token management service is a service of the trusted data management service.

5. The method of claim 1, wherein the query to the digital rights management service comprises a key identifier and a key encryption key.

6. The method of claim 5, wherein the key identifier and the key encryption key are accessed from a database associated with a trusted data management service.

7. The method of claim 6, wherein the key encryption key comprises a public encryption key associated with the user.

8. The method of claim 1, wherein the digital rights management token comprises an encrypted content key.

9. The method of claim 1, wherein the location of the digital asset comprises a uniform resource locator associated with an electronic file corresponding to the digital asset.

10. The method of claim 1, wherein determining that the user has rights to the digital asset comprises determining that the user is an owner of the digital asset.

11. The method of claim 10, wherein the method further comprises verifying, by the token rights management service, that there is remaining supply of the digital asset.

12. The method of claim 1, wherein the digital rights management data access request comprises at least one of an identifier associated with the digital asset, an identifier associated with a user that issued the user request to the marketplace service, and an identifier associated with a product that includes the digital asset.

13. The method of claim 1, wherein the marketplace service executes on the same system as the token rights management service.

14. The method of claim 1, wherein the trusted ledger comprises a trusted immutable distributed assertion ledger.

15. The method of claim 1, wherein the trusted ledger comprises a blockchain ledger.

16. The method of claim 1, wherein the digital rights management service is a service of the trusted data management service.

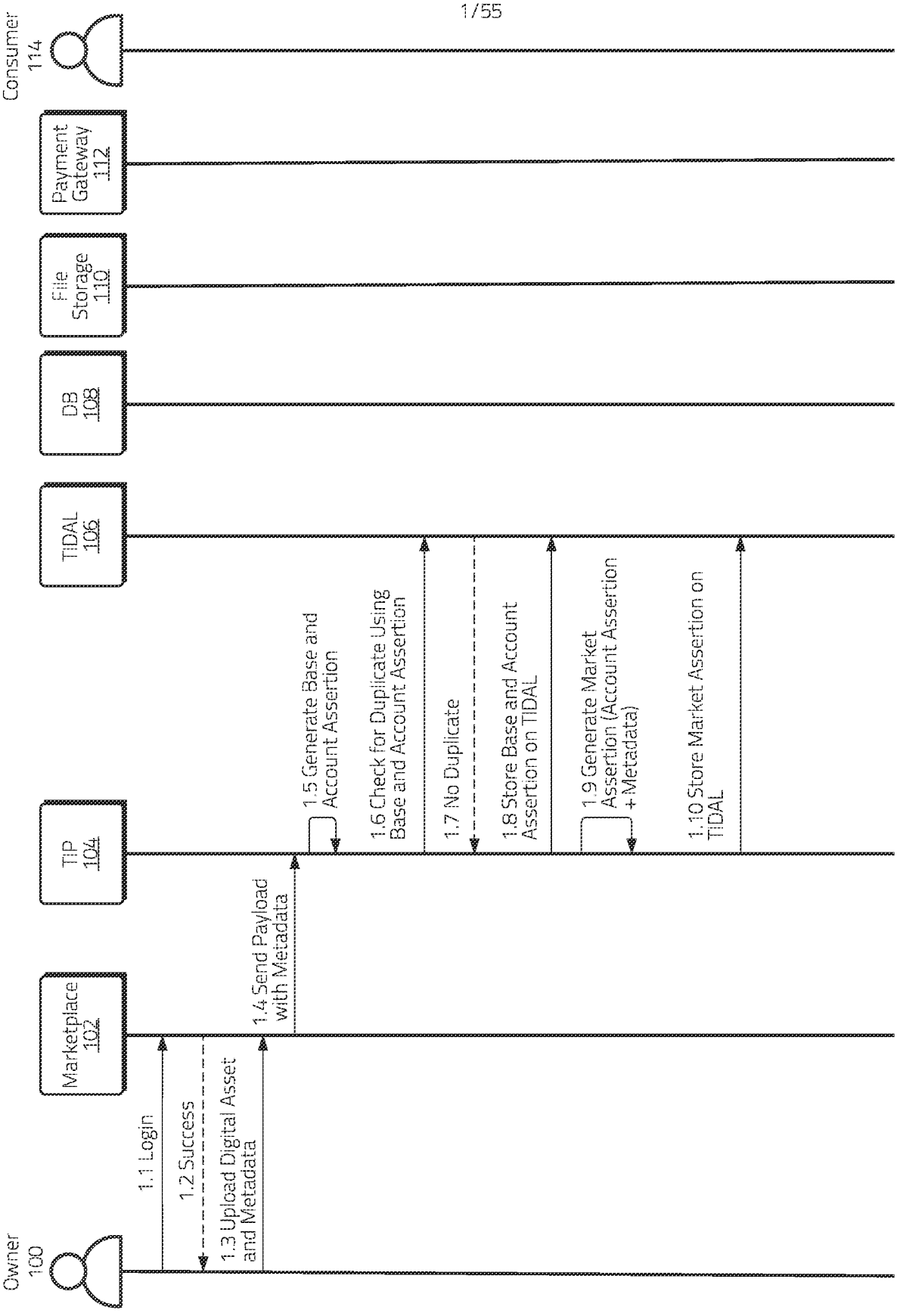

Owner
100
Consumer
114
Marketplace
102
TIP
104
TIDAL
106
DB
108
File
Storage
110
Payment
Gateway
112
1.1 Login
1.2 Success
1.3 Upload Digital Asset and Metadata
1.4 Send Payload with Metadata
1.5 Generate Base and Account Assertion
1.6 Check for Duplicate Using Base and Account Assertion
1.7 No Duplicate
1.8 Store Base and Account Assertion on TIDAL
1.9 Generate Market Assertion (Account Assertion + Metadata)
1.10 Store Market Assertion on TIDAL

Figure 1A

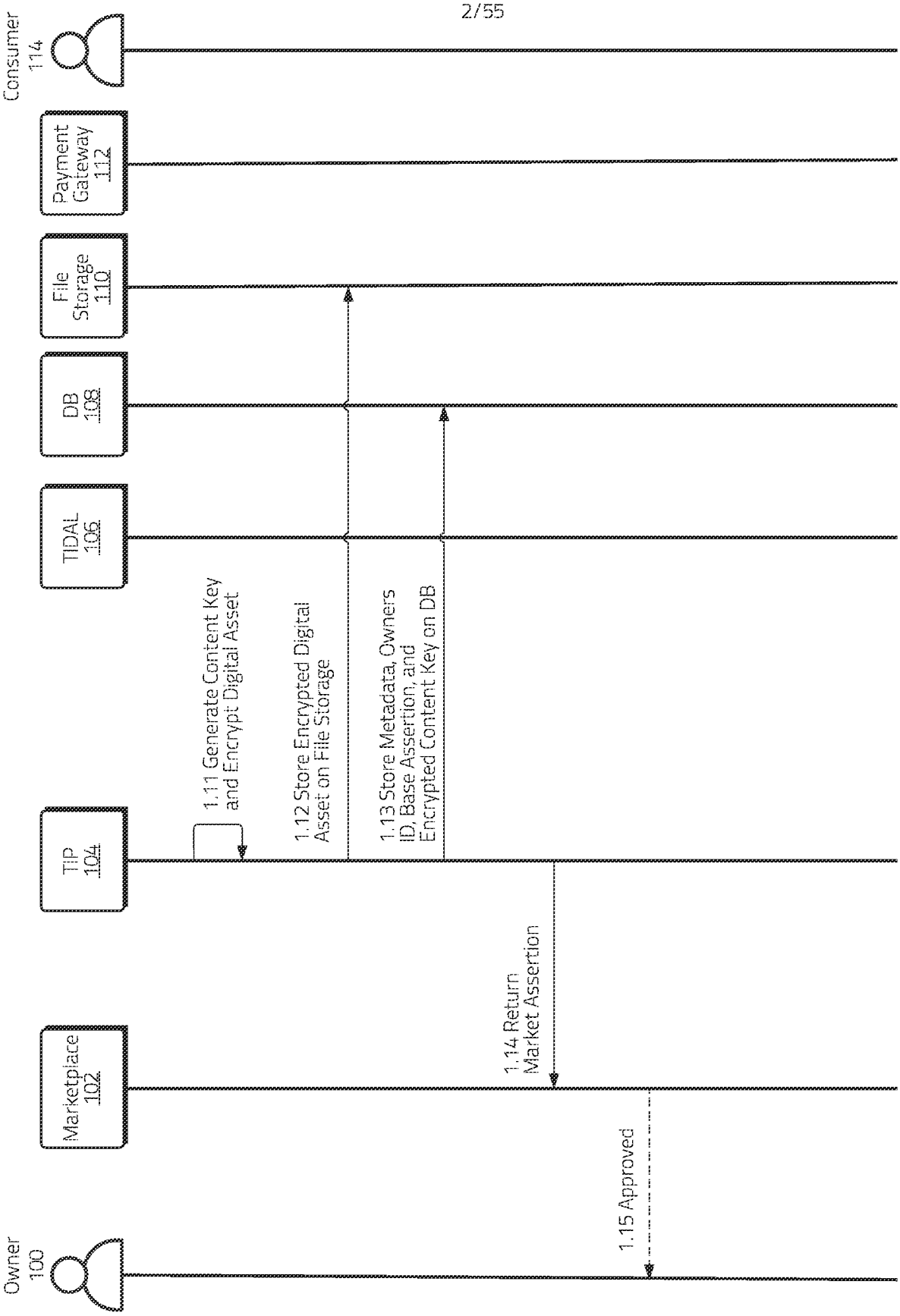

Consumer 114
Payment Gateway 112
File Storage 110
DB 108
TIDAL 106
TIP 104
Marketplace 102
Owner 100
1.11 Generate Content Key and Encrypt Digital Asset
1.12 Store Encrypted Digital Asset on File Storage
1.13 Store Metadata, Owners ID, Base Assertion, and Encrypted Content Key on DB
1.14 Return Market Assertion
1.15 Approved

Figure 1B

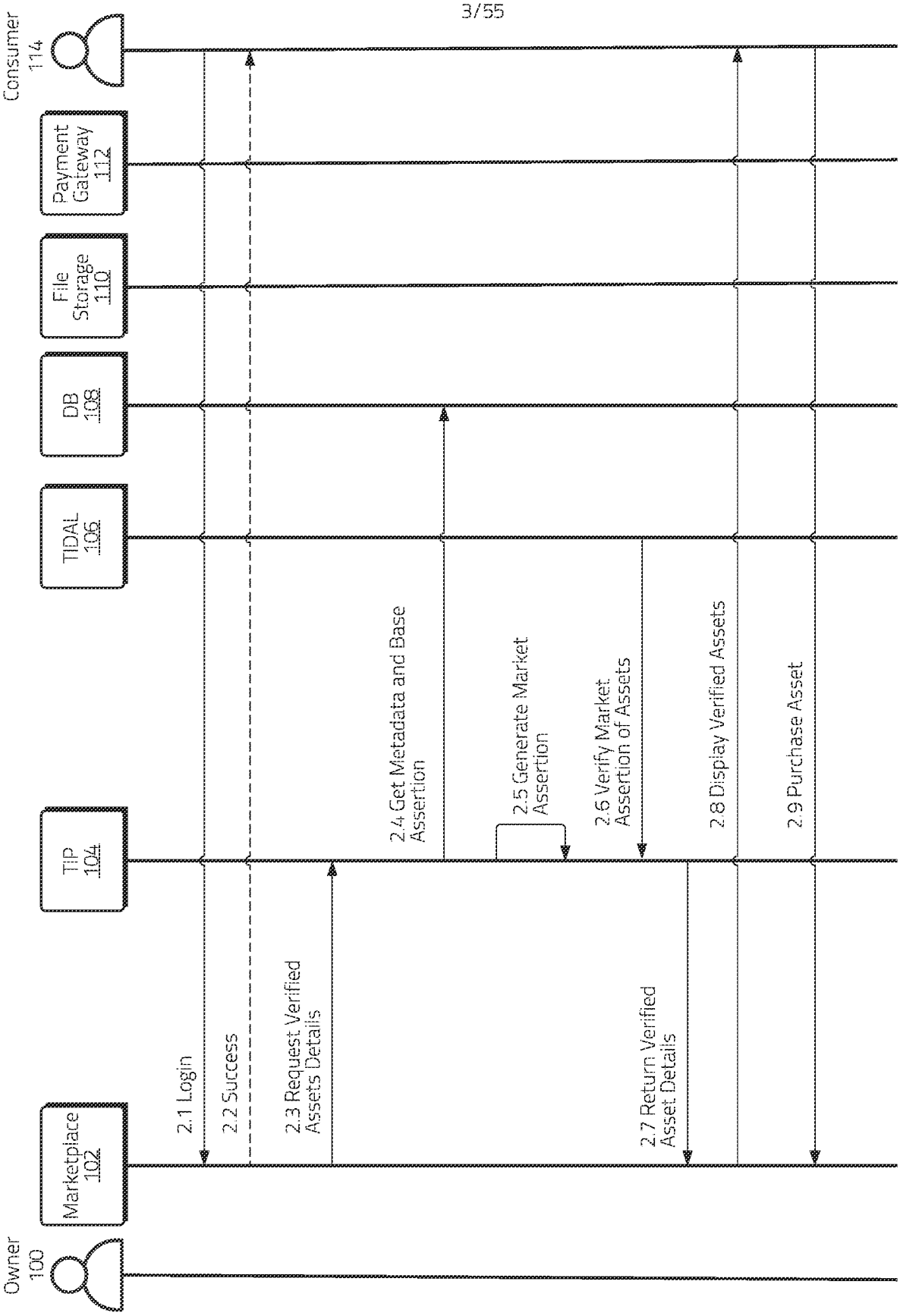

Owner
100
Consumer
114
Marketplace
102
TIP
104
TIDAL
106
DB
108
File
Storage
110
Payment
Gateway
112
2.1 Login
2.2 Success
2.3 Request Verified
Assets Details
2.4 Get Metadata and Base
Assertion
2.5 Generate Market
Assertion
2.6 Verify Market
Assertion of Assets
2.7 Return Verified
Asset Details
2.8 Display Verified Assets
2.9 Purchase Asset

Figure 2A

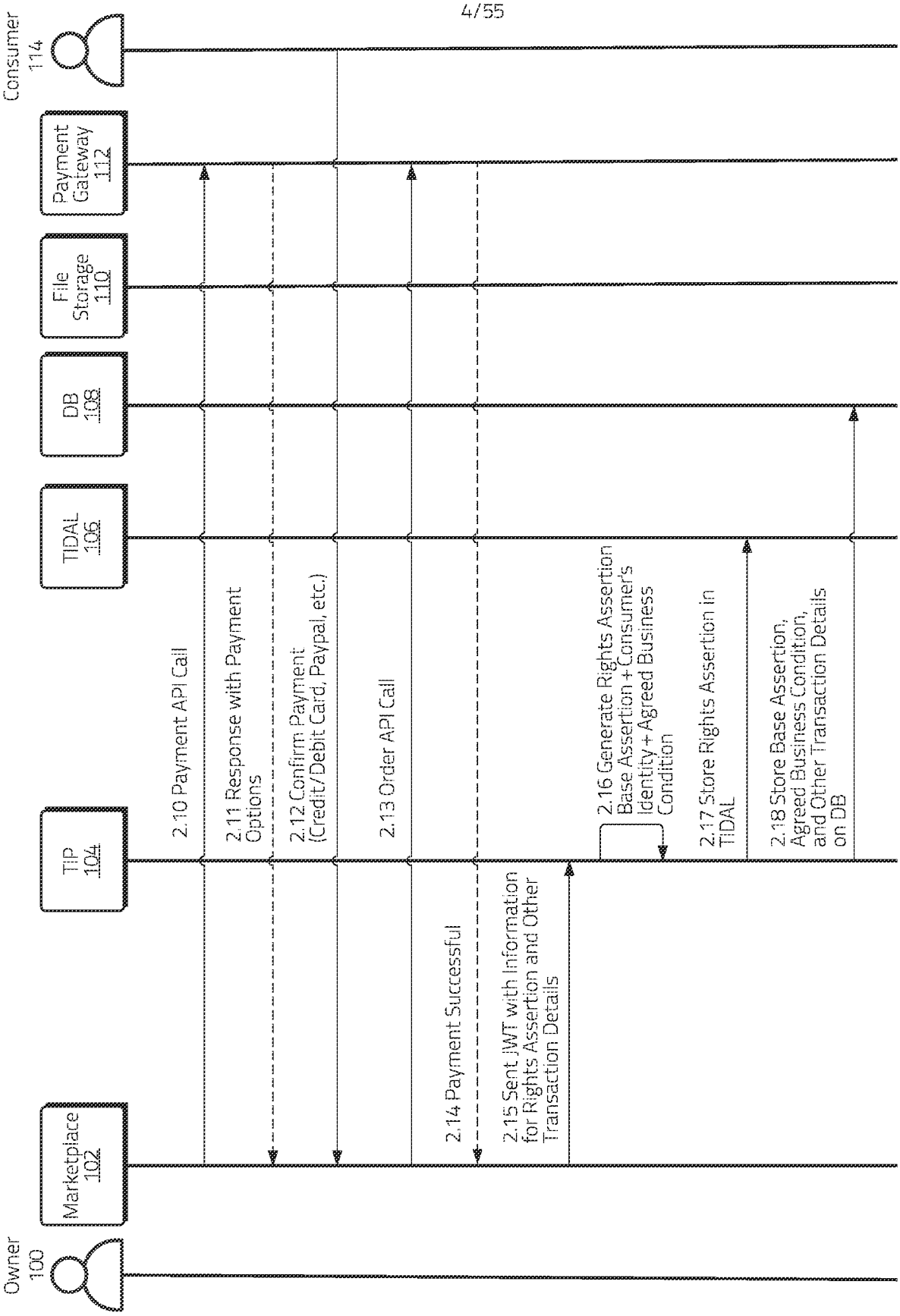

Owner
100
Consumer
114
Marketplace
102
TIP
104
TIDAL
106
DB
108
File
Storage
110
Payment
Gateway
112
2.10 Payment API Call
2.11 Response with Payment Options
2.12 Confirm Payment (Credit/Debit Card, Paypal, etc.)
2.13 Order API Call
2.14 Payment Successful
2.15 Sent JWT with Information for Rights Assertion and Other Transaction Details
2.16 Generate Rights Assertion Base Assertion + Consumer's Identity + Agreed Business Condition
2.17 Store Rights Assertion in TIDAL
2.18 Store Base Assertion, Agreed Business Condition, and Other Transaction Details on DB

Figure 2B

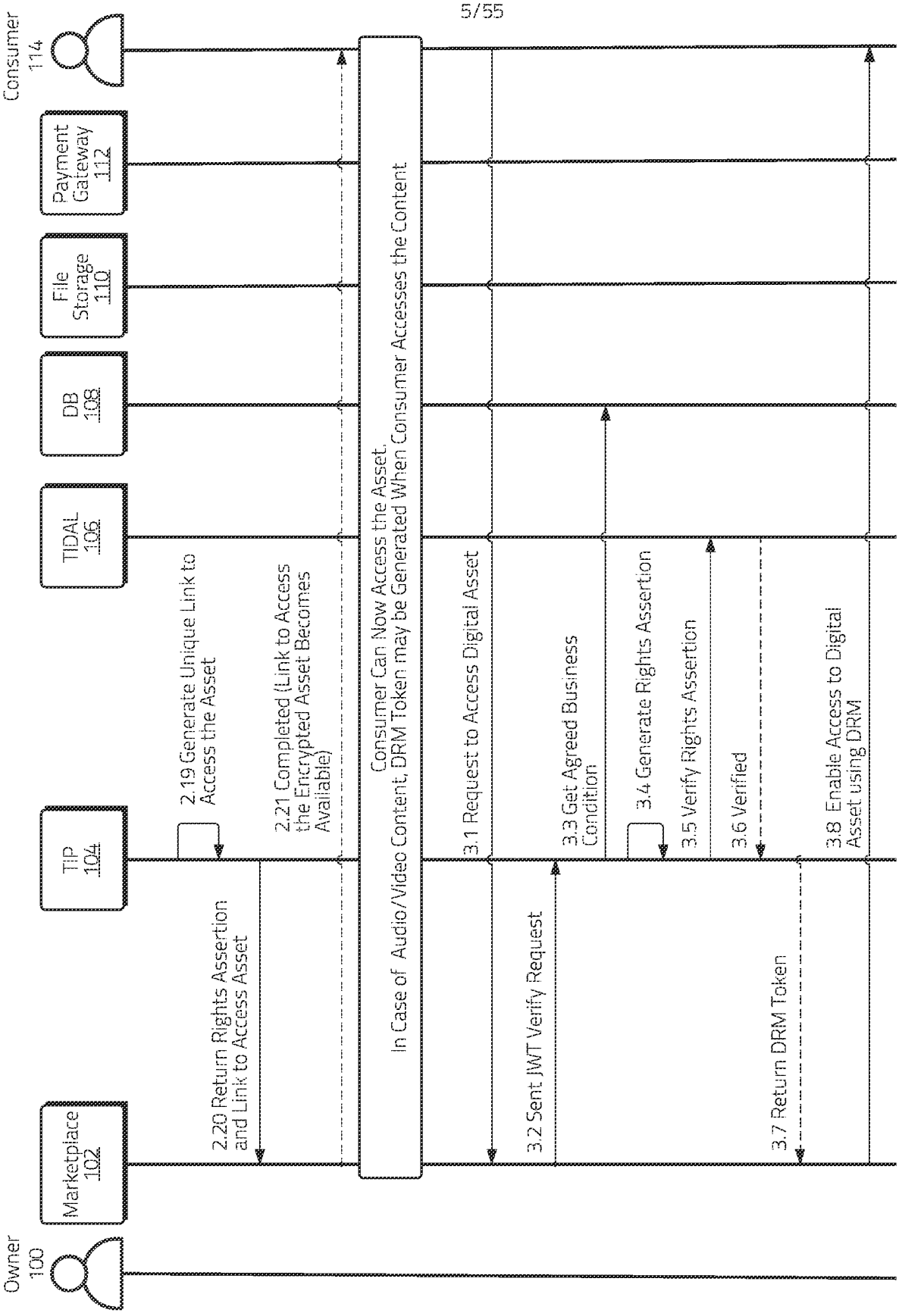

Owner
100
Marketplace
102
TIP
104
TIDAL
106
DB
108
File
Storage
110
Payment
Gateway
112
Consumer
114
2.19 Generate Unique Link to Access the Asset
2.20 Return Rights Assertion and Link to Access Asset
2.21 Completed (Link to Access the Encrypted Asset Becomes Available)
Consumer Can Now Access the Asset,
In Case of Audio/Video Content, DRM Token may be Generated When Consumer Accesses the Content
3.1 Request to Access Digital Asset
3.2 Sent JWT Verify Request
3.3 Get Agreed Business Condition
3.4 Generate Rights Assertion
3.5 Verify Rights Assertion
3.6 Verified
3.7 Return DRM Token
3.8 Enable Access to Digital Asset using DRM

Figure 2C

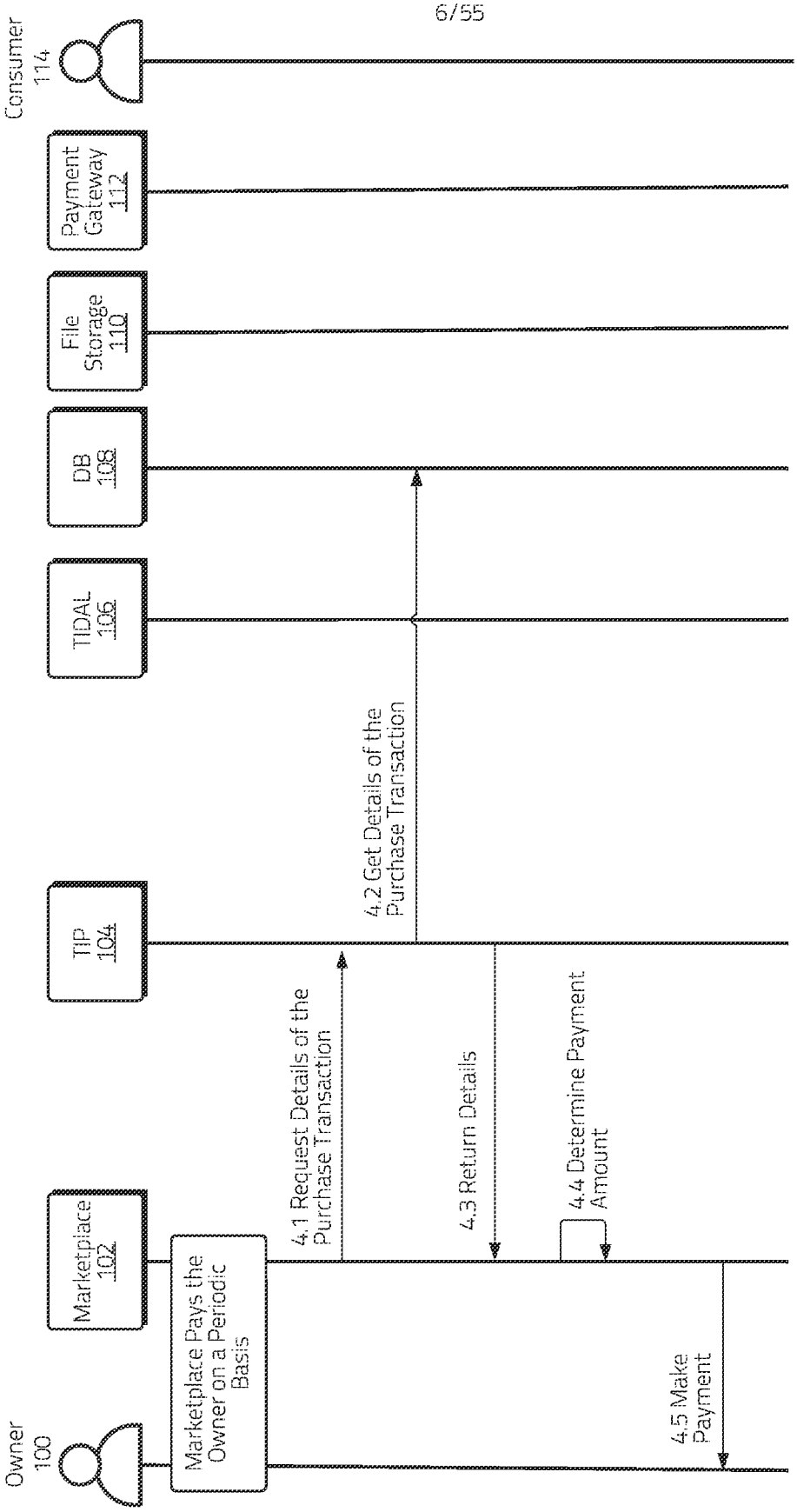

Owner
100
Consumer
114
Marketplace
102
TIP
104
TIDAL
106
DB
108
File
Storage
110
Payment
Gateway
112
Marketplace Pays the Owner on a Periodic Basis
4.1 Request Details of the Purchase Transaction
4.2 Get Details of the Purchase Transaction
4.3 Return Details
4.4 Determine Payment Amount
4.5 Make Payment

Figure 3

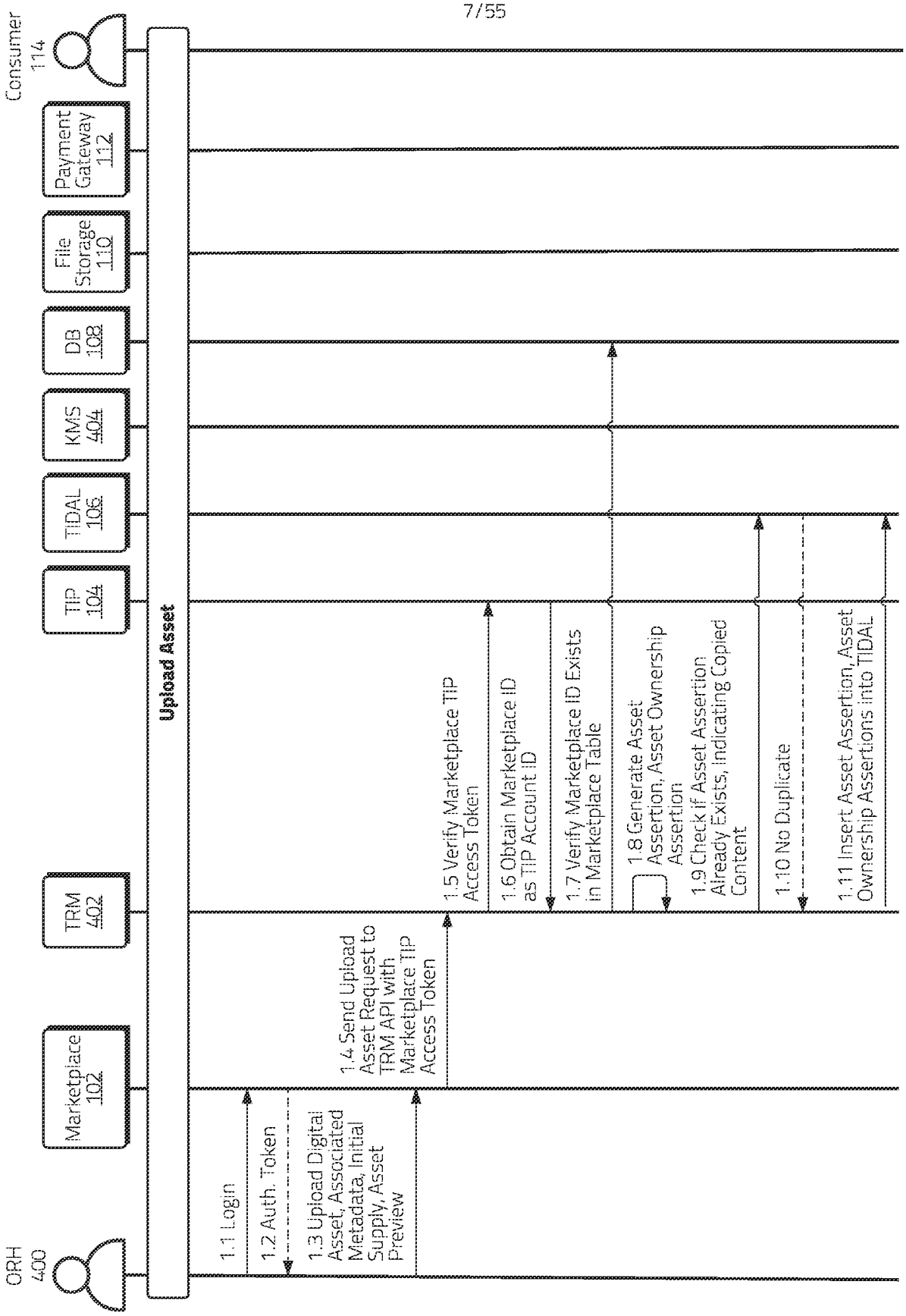

ORH 400
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Upload Asset
1.1 Login
1.2 Auth. Token
1.3 Upload Digital Asset, Associated Metadata, Initial Supply, Asset Preview
1.4 Send Upload Asset Request to TRM API with Marketplace TIP Access Token
1.5 Verify Marketplace TIP Access Token
1.6 Obtain Marketplace ID as TIP Account ID
1.7 Verify Marketplace ID Exists in Marketplace Table
1.8 Generate Asset Assertion, Asset Ownership Assertion
1.9 Check if Asset Assertion Already Exists, Indicating Copied Content
1.10 No Duplicate
1.11 Insert Asset Assertion, Asset Ownership Assertions into TIDAL

Figure 4A

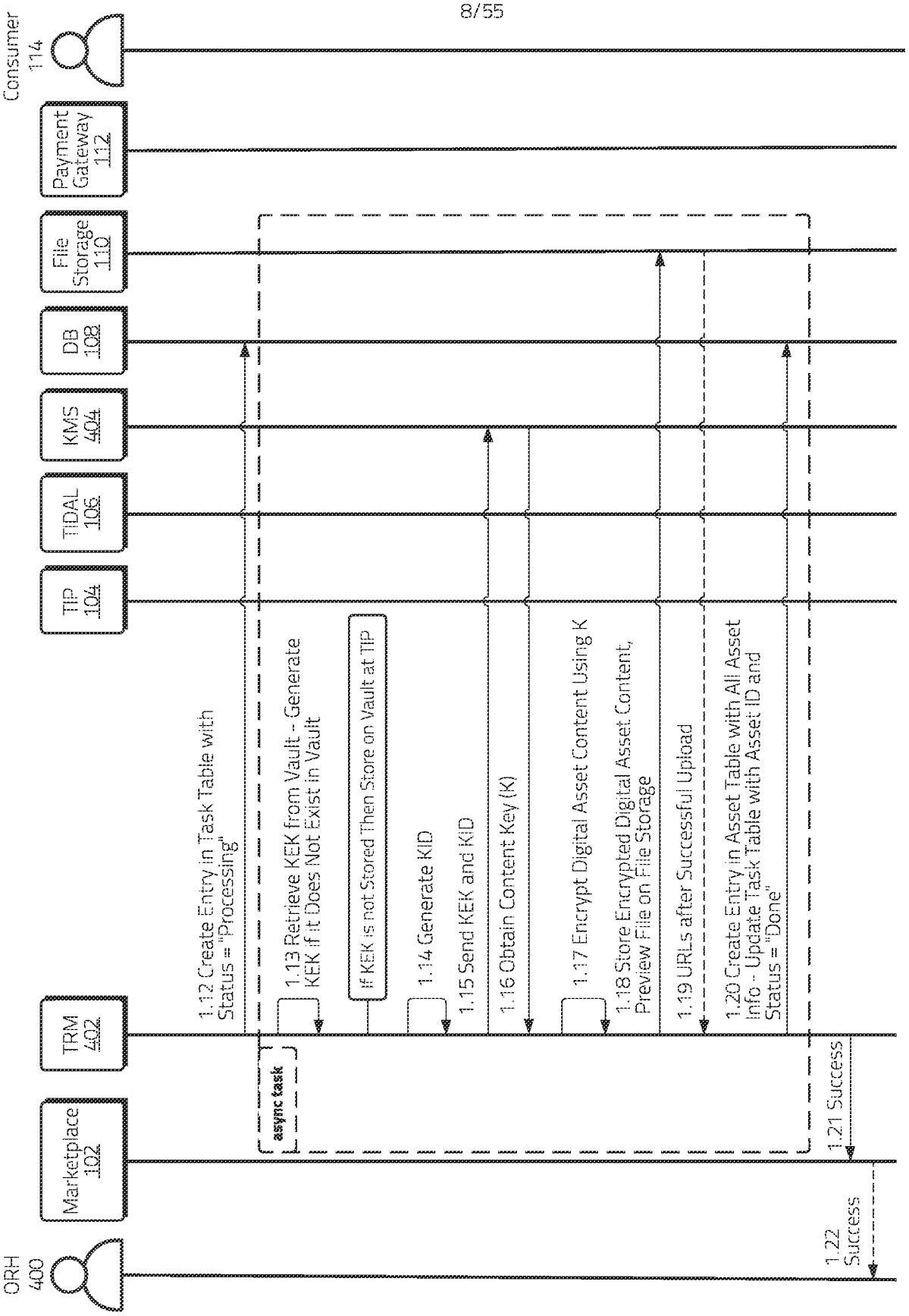

ORH 400
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
async task
1.12 Create Entry in Task Table with Status = "Processing"
1.13 Retrieve KEK from Vault - Generate KEK if it Does Not Exist in Vault
If KEK is not Stored Then Store on Vault at TIP
1.14 Generate KID
1.15 Send KEK and KID
1.16 Obtain Content Key (K)
1.17 Encrypt Digital Asset Content Using K
1.18 Store Encrypted Digital Asset Content, Preview File on File Storage
1.19 URLs after Successful Upload
1.20 Create Entry in Asset Table with All Asset Info - Update Task Table with Asset ID and Status = "Done"
1.21 Success
1.22 Success

Figure 4B

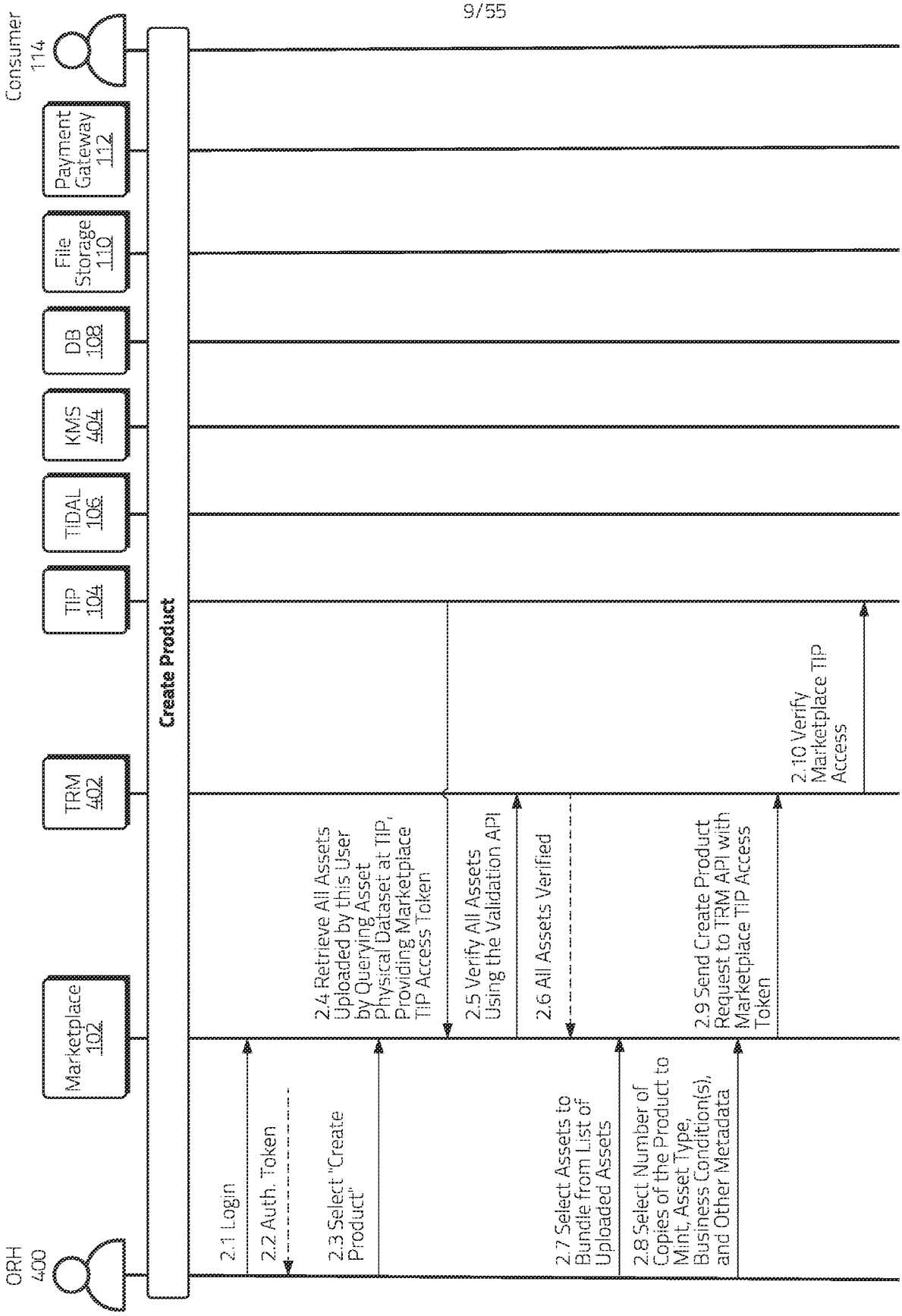

ORH 400
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
Create Product
2.1 Login
2.2 Auth. Token
2.3 Select "Create Product"
2.4 Retrieve All Assets Uploaded by this User by Querying Asset Physical Dataset at TIP, Providing Marketplace TIP Access Token
2.5 Verify All Assets Using the Validation API
2.6 All Assets Verified
2.7 Select Assets to Bundle from List of Uploaded Assets
2.8 Select Number of Copies of the Product to Mint, Asset Type, Business Condition(s), and Other Metadata
2.9 Send Create Product Request to TRM API with Marketplace TIP Access Token
2.10 Verify Marketplace TIP Access

Figure 5A

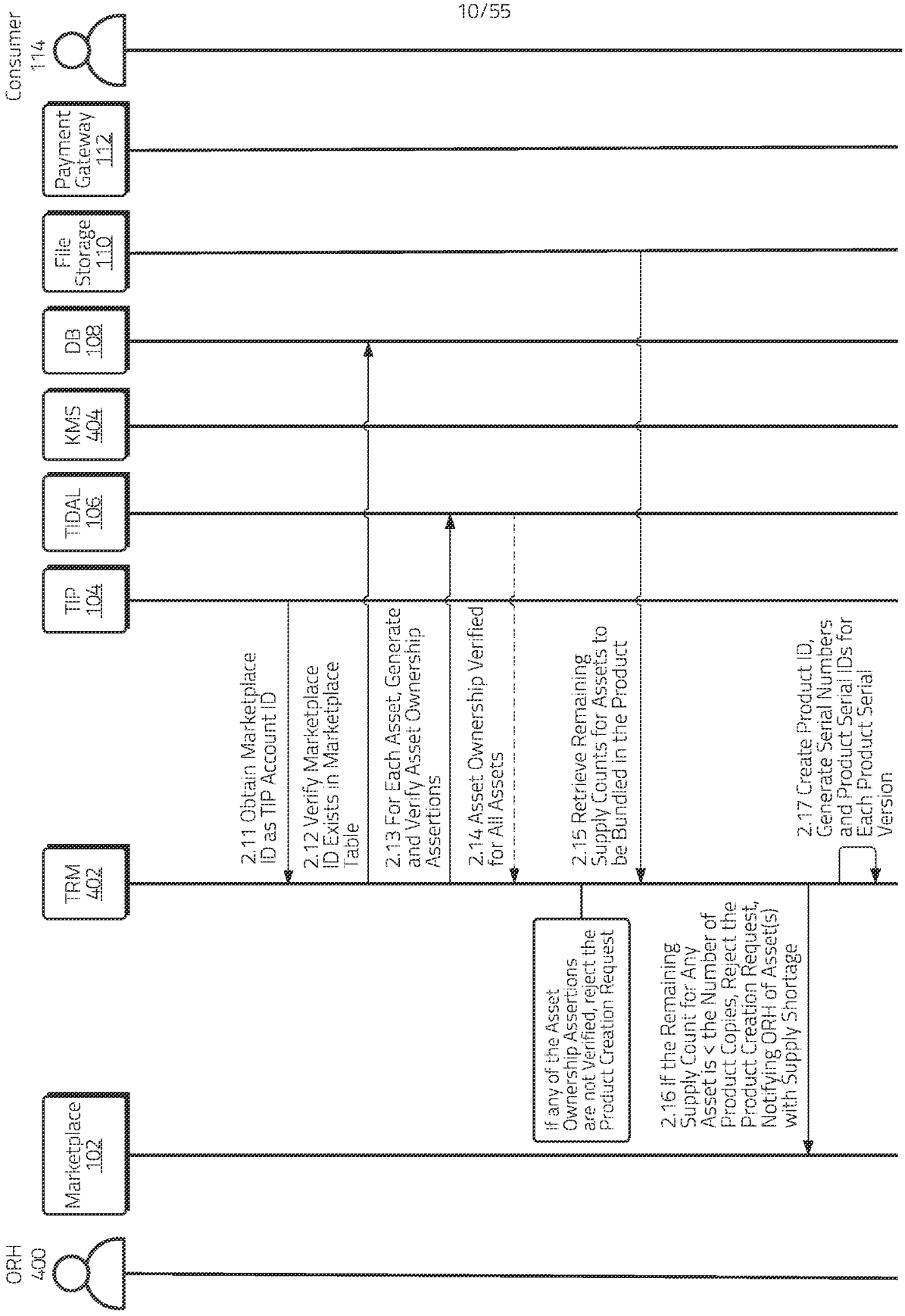

ORH 400
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
2.11 Obtain Marketplace ID as TIP Account ID
2.12 Verify Marketplace ID Exists in Marketplace Table
2.13 For Each Asset, Generate and Verify Asset Ownership Assertions
2.14 Asset Ownership Verified for All Assets
2.15 Retrieve Remaining Supply Counts for Assets to be Bundled in the Product
2.17 Create Product ID, Generate Serial Numbers and Product Serial IDs for Each Product Serial Version
If any of the Asset Ownership Assertions are not Verified, reject the Product Creation Request
2.16 If the Remaining Supply Count for Any Asset is < the Number of Product Copies, Reject the Product Creation Request, Notifying ORH of Asset(s) with Supply Shortage

Figure 5B

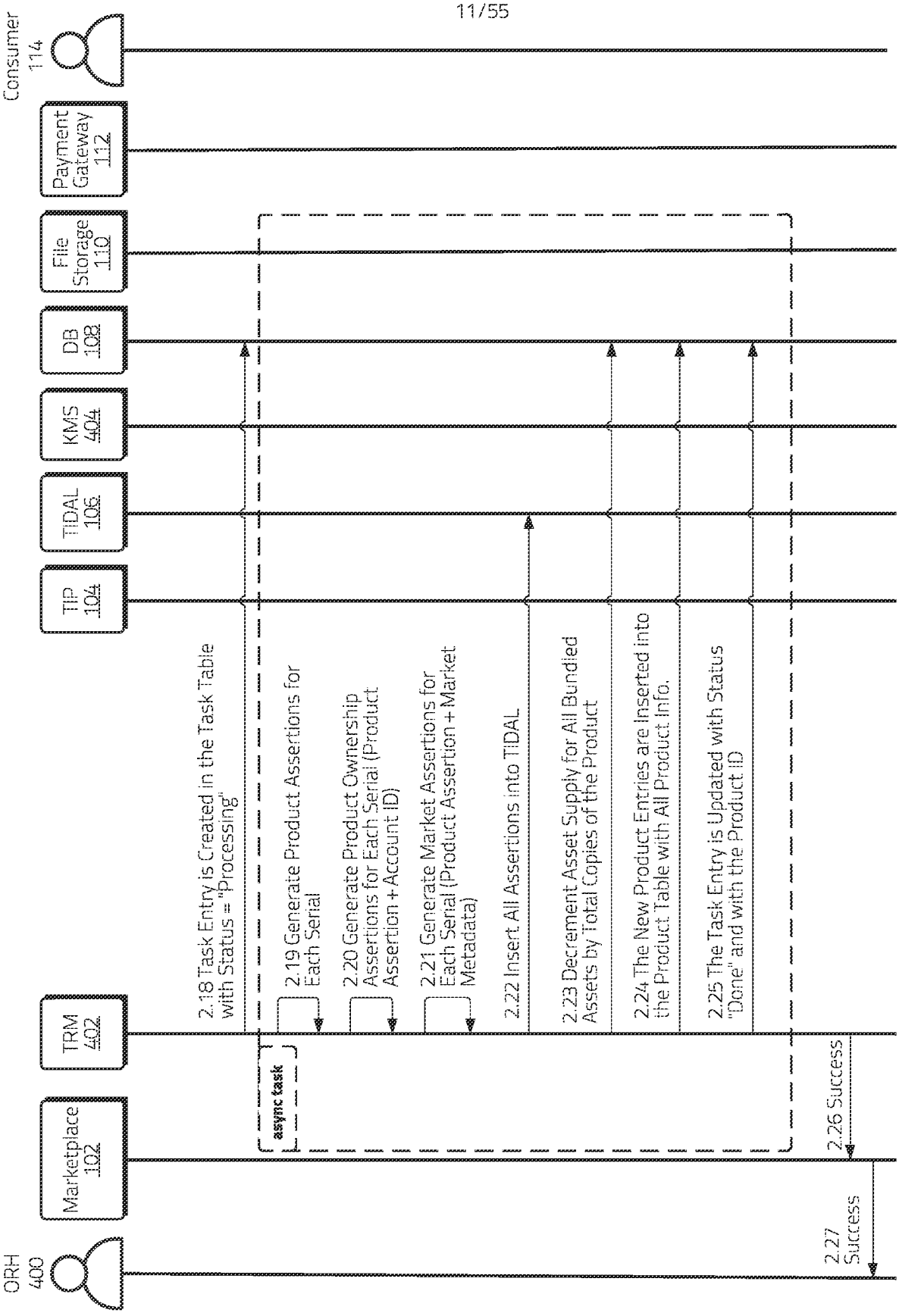

ORH 400
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
2.18 Task Entry is Created in the Task Table with Status = "Processing"
async task
2.19 Generate Product Assertions for Each Serial
2.20 Generate Product Ownership Assertions for Each Serial (Product Assertion + Account ID)
2.21 Generate Market Assertions for Each Serial (Product Assertion + Market Metadata)
2.22 Insert All Assertions into TIDAL
2.23 Decrement Asset Supply for All Bundled Assets by Total Copies of the Product
2.24 The New Product Entries are Inserted into the Product Table with All Product Info.
2.25 The Task Entry is Updated with Status "Done" and with the Product ID
2.26 Success
2.27 Success

Figure 5C

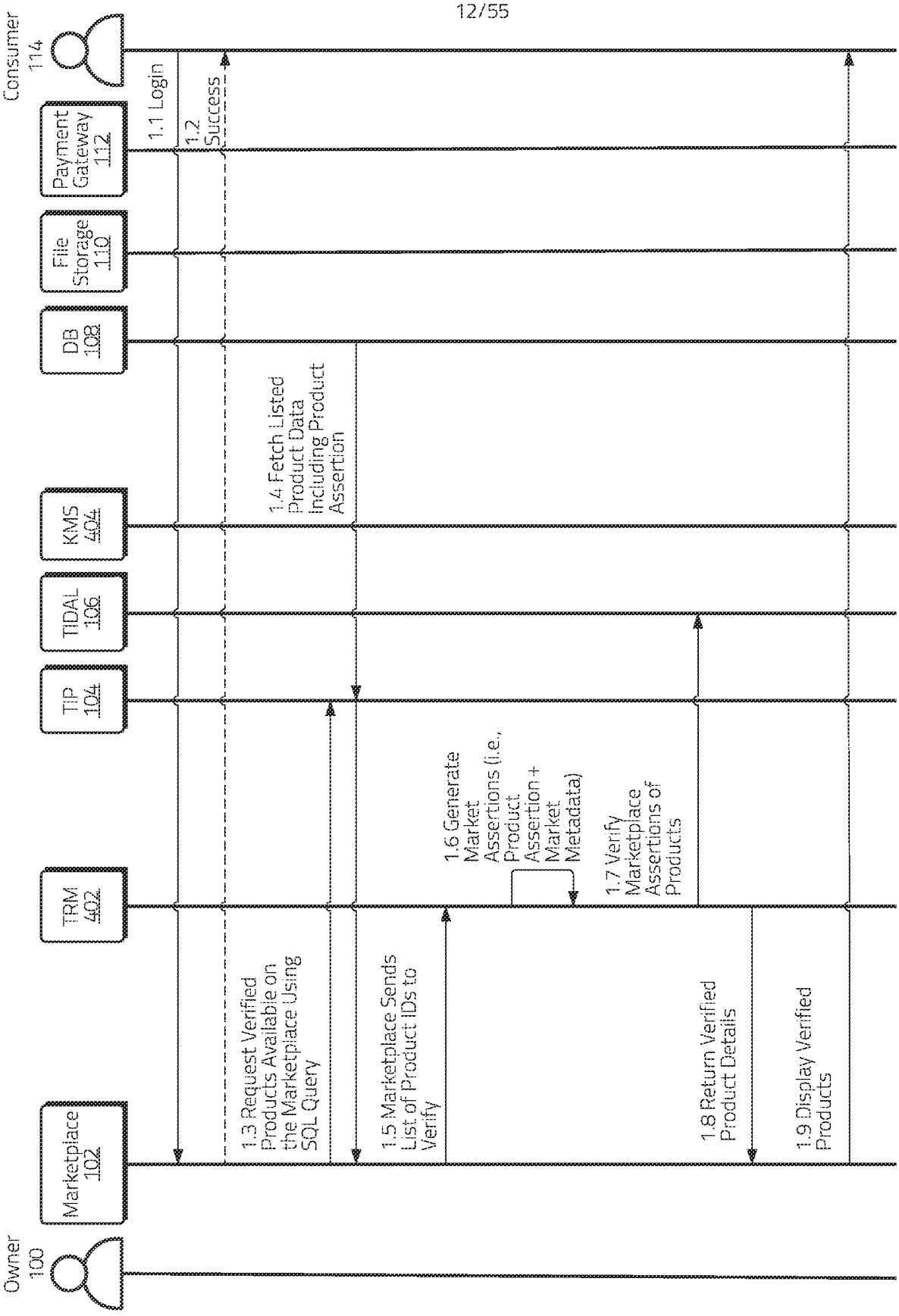

Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
1.1 Login
1.2 Success
1.3 Request Verified Products Available on the Marketplace Using SQL Query
1.4 Fetch Listed Product Data Including Product Assertion
1.5 Marketplace Sends List of Product IDs to Verify
1.6 Generate Market Assertions (i.e., Product Assertion + Market Metadata)
1.7 Verify Marketplace Assertions of Products
1.8 Return Verified Product Details
1.9 Display Verified Products

Figure 6A

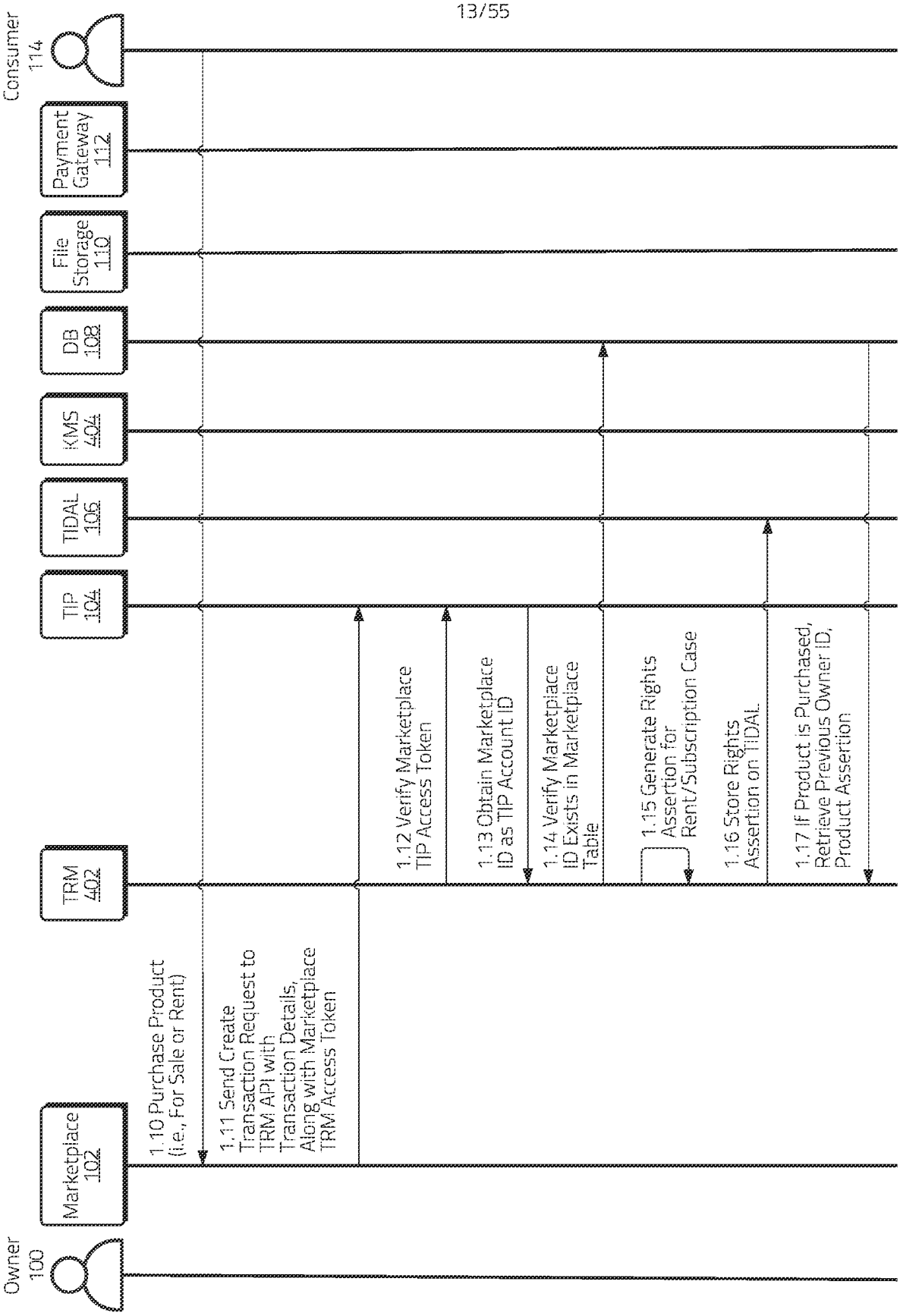

Owner 100
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
1.10 Purchase Product (i.e., For Sale or Rent)
1.11 Send Create Transaction Request to TRM API with Transaction Details, Along with Marketplace TRM Access Token
1.12 Verify Marketplace TIP Access Token
1.13 Obtain Marketplace ID as TIP Account ID
1.14 Verify Marketplace ID Exists in Marketplace Table
1.15 Generate Rights Assertion for Rent/Subscription Case
1.16 Store Rights Assertion on TIDAL
1.17 If Product is Purchased, Retrieve Previous Owner ID, Product Assertion

Figure 6B

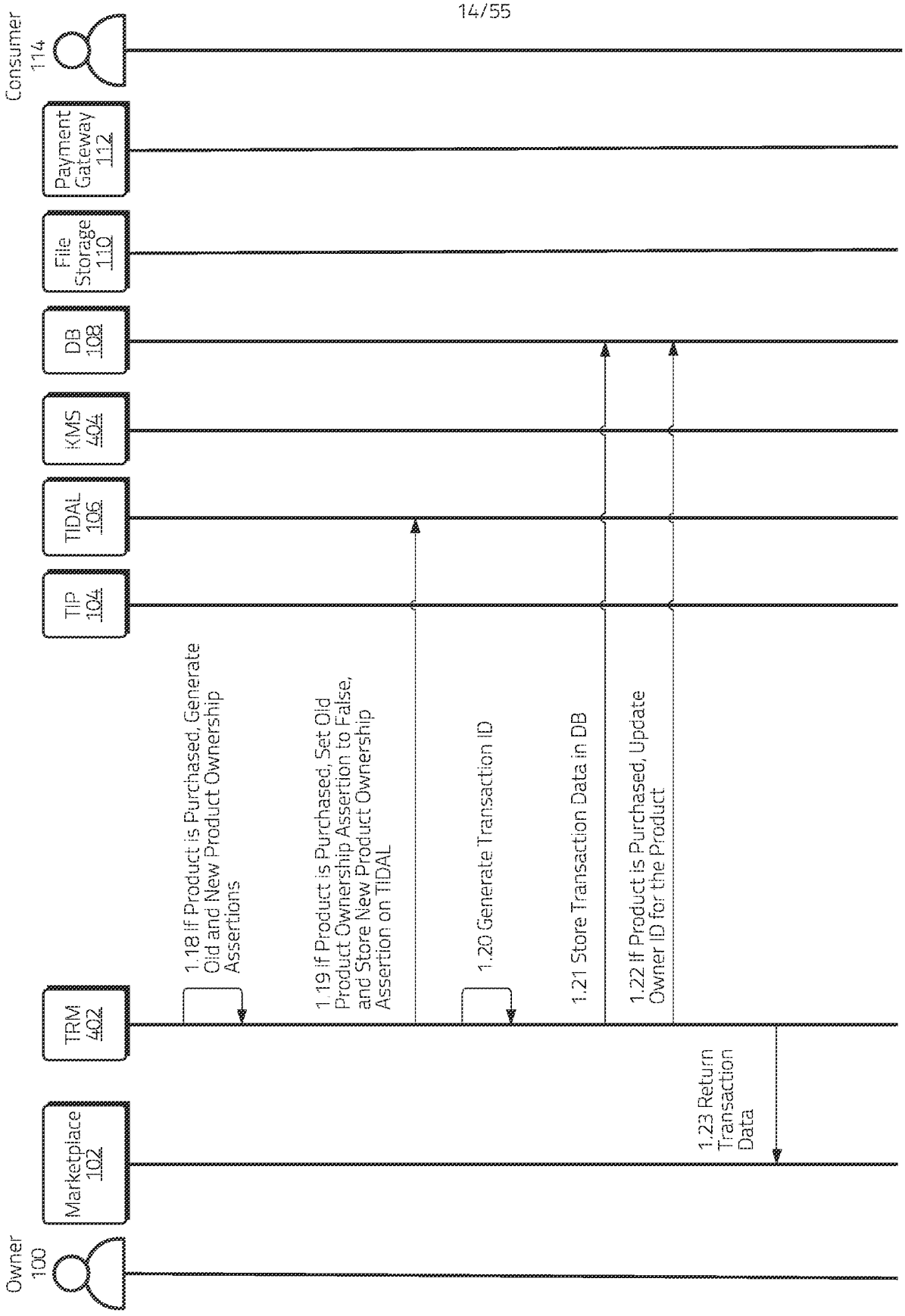

Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
1.18 If Product is Purchased, Generate Old and New Product Ownership Assertions
1.19 If Product is Purchased, Set Old Product Ownership Assertion to False, and Store New Product Ownership Assertion on TIDAL
1.20 Generate Transaction ID
1.21 Store Transaction Data in DB
1.22 If Product is Purchased, Update Owner ID for the Product
1.23 Return Transaction Data

Figure 6C

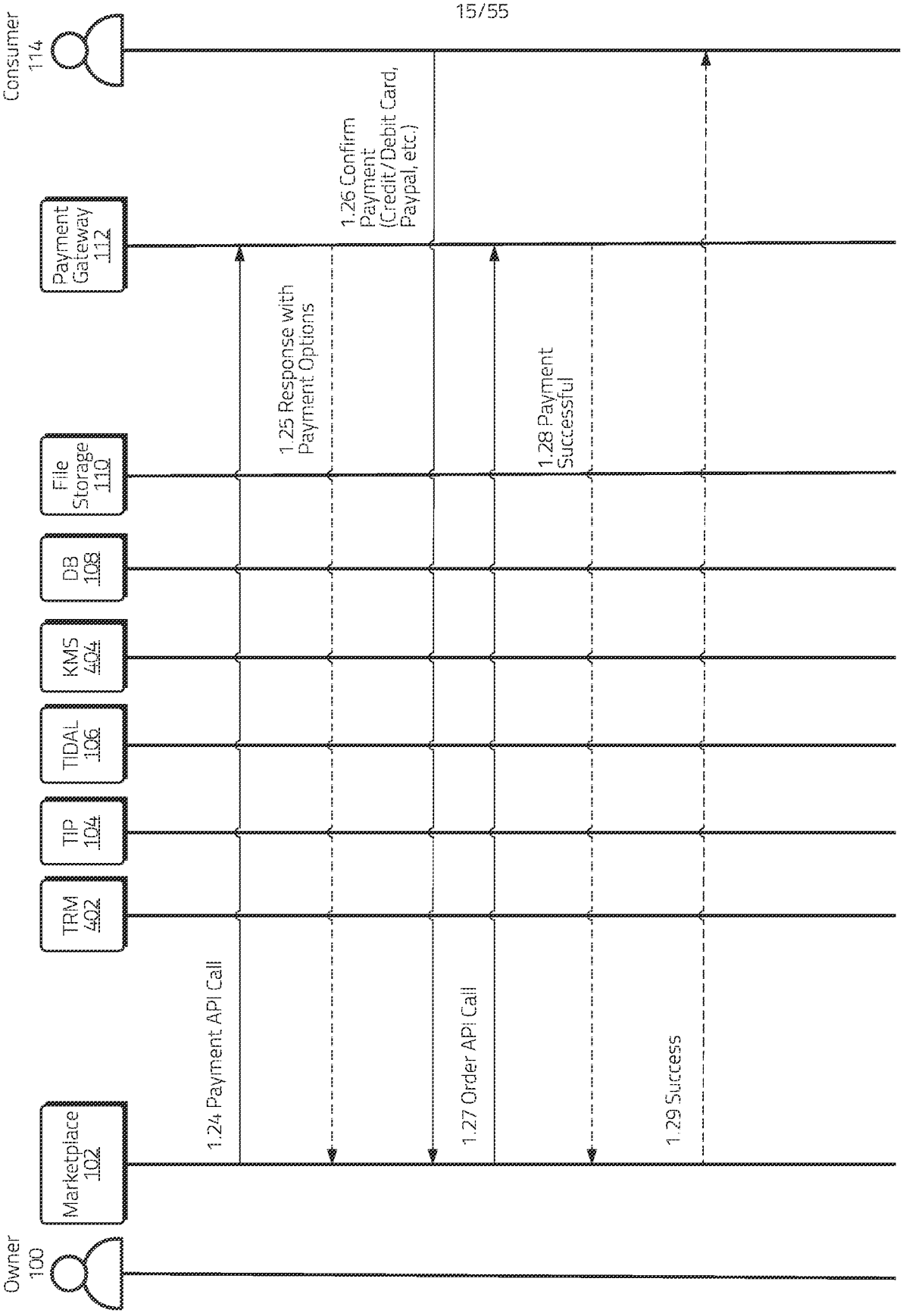

Owner 100
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
1.24 Payment API Call
1.25 Response with Payment Options
1.26 Confirm Payment (Credit/Debit Card, Paypal, etc.)
1.27 Order API Call
1.28 Payment Successful
1.29 Success

Figure 6D

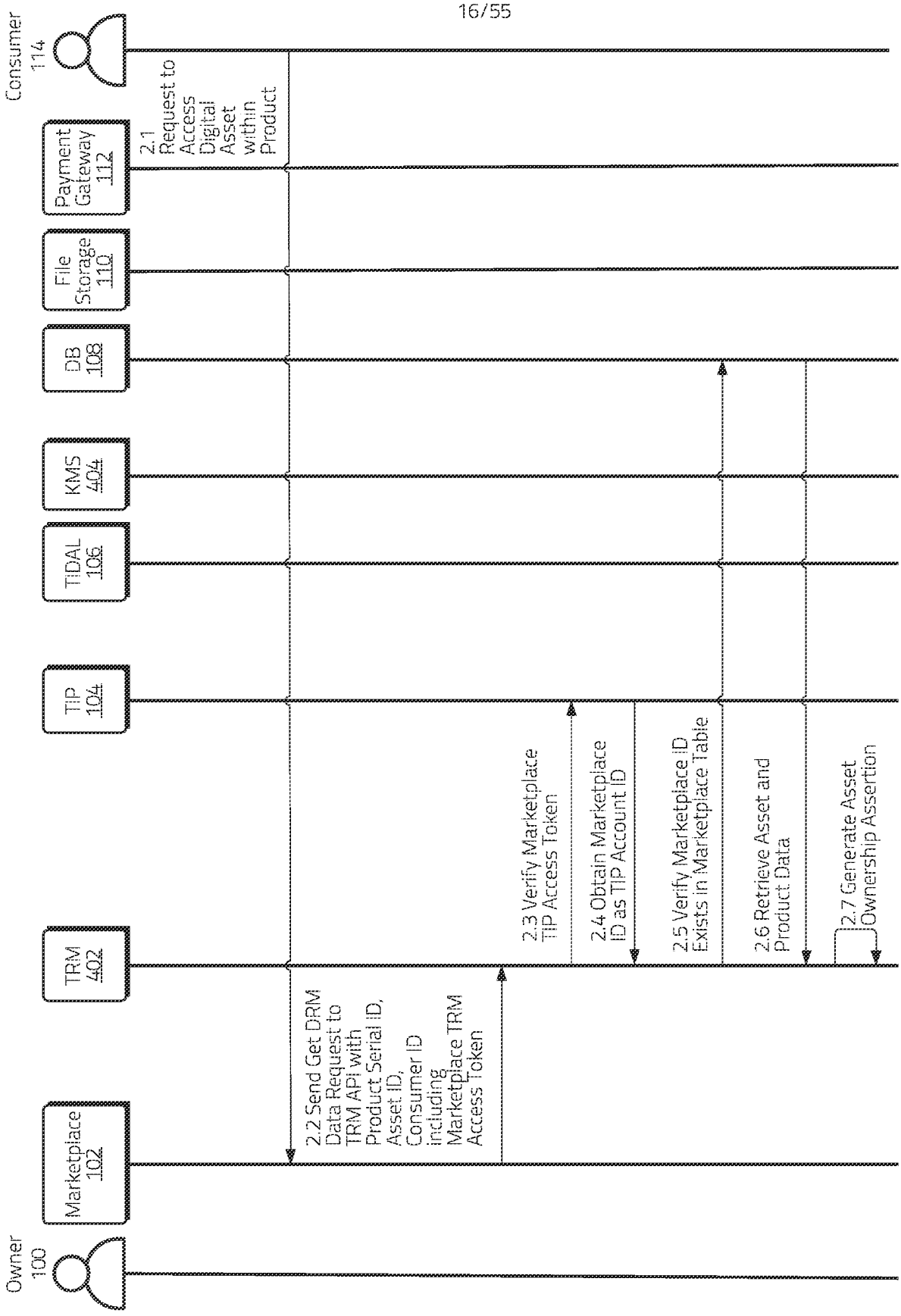

Owner
100
Consumer
114
Marketplace
102
TRM
402
TIP
104
TIDAL
106
KMS
404
DB
108
File
Storage
110
Payment
Gateway
112
2.1 Request to Access Digital Asset within Product
2.2 Send Get DRM Data Request to TRM API with Product Serial ID, Asset ID, Consumer ID including Marketplace TRM Access Token
2.3 Verify Marketplace TIP Access Token
2.4 Obtain Marketplace ID as TIP Account ID
2.5 Verify Marketplace ID Exists in Marketplace Table
2.6 Retrieve Asset and Product Data
2.7 Generate Asset Ownership Assertion

Figure 7A

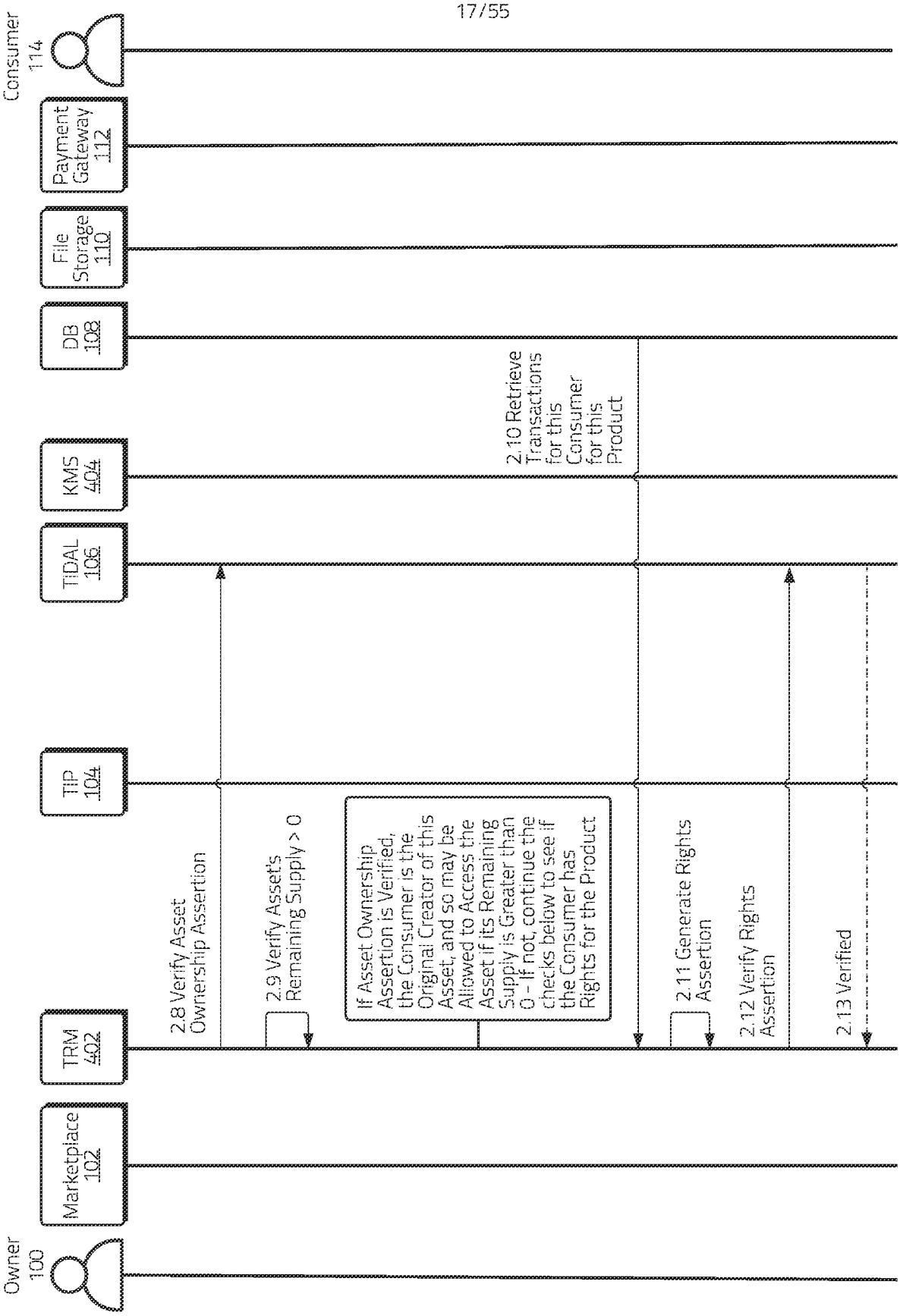

Owner 100
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
2.8 Verify Asset Ownership Assertion
2.9 Verify Asset's Remaining Supply > 0
If Asset Ownership Assertion is Verified, the Consumer is the Original Creator of this Asset, and so may be Allowed to Access the Asset if its Remaining Supply is Greater than 0 - If not, continue the checks below to see if the Consumer has Rights for the Product
2.10 Retrieve Transactions for this Consumer for this Product
2.11 Generate Rights Assertion
2.12 Verify Rights Assertion
2.13 Verified

Figure 7B

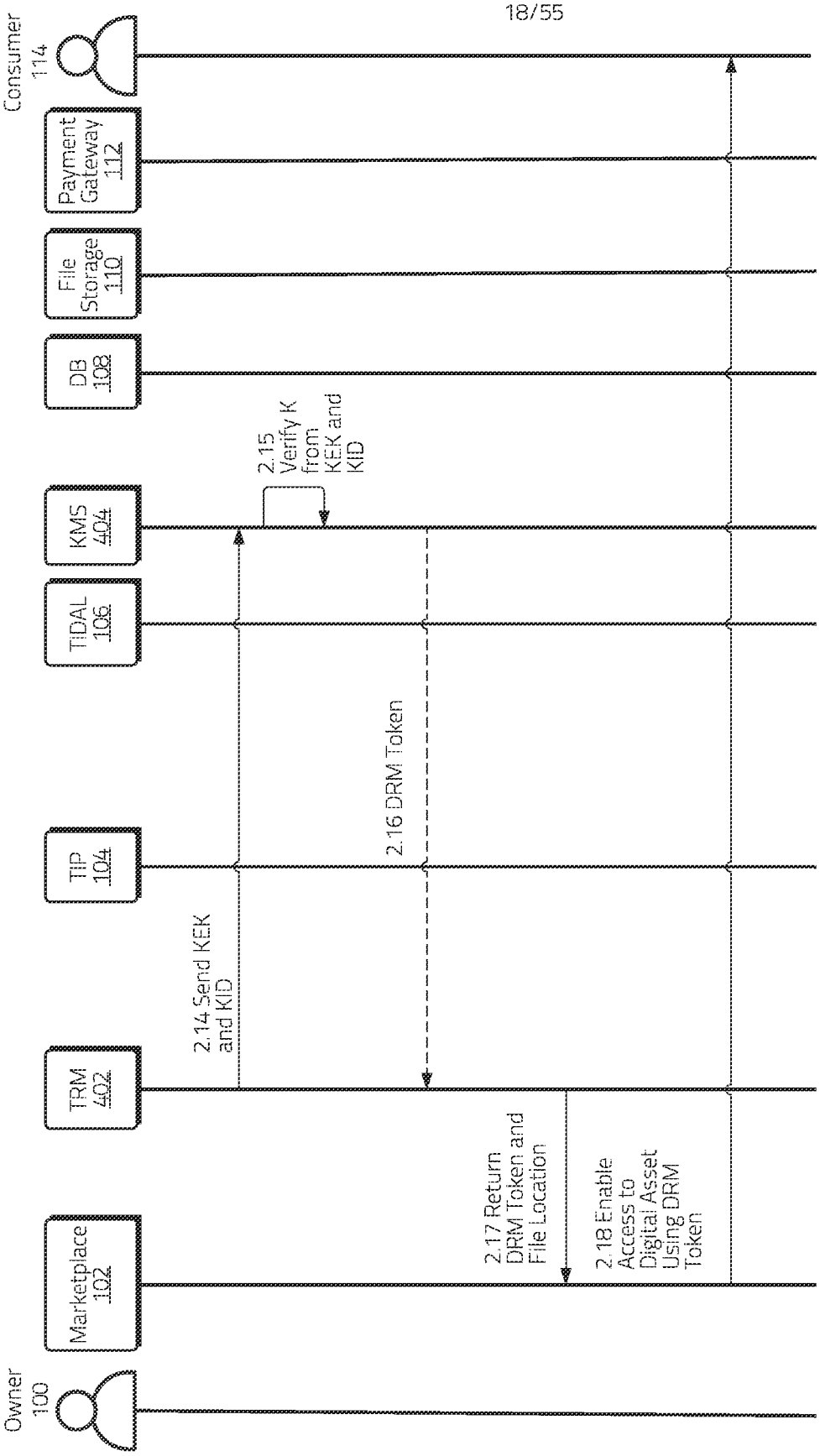

Owner 100
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
2.14 Send KEK and KID
2.15 Verify K from KEK and KID
2.16 DRM Token
2.17 Return DRM Token and File Location
2.18 Enable Access to Digital Asset Using DRM Token

Figure 7C

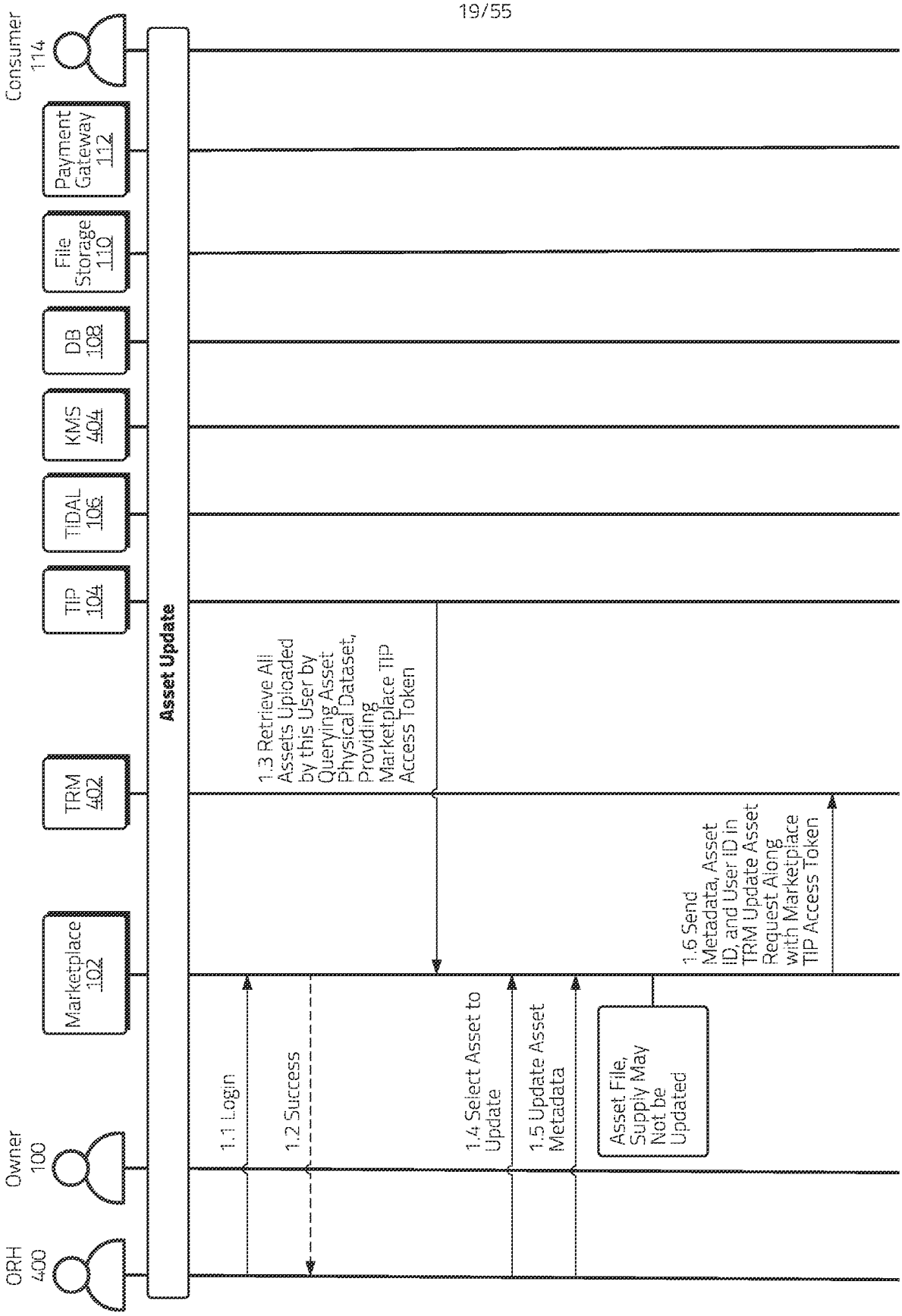

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Asset Update
1.1 Login
1.2 Success
1.3 Retrieve All Assets Uploaded by this User by Querying Asset Physical Dataset, Providing Marketplace TIP Access Token
1.4 Select Asset to Update
1.5 Update Asset Metadata
Asset File, Supply May Not be Updated
1.6 Send Metadata, Asset ID, and User ID in TRM Update Asset Request Along with Marketplace TIP Access Token

Figure 8A

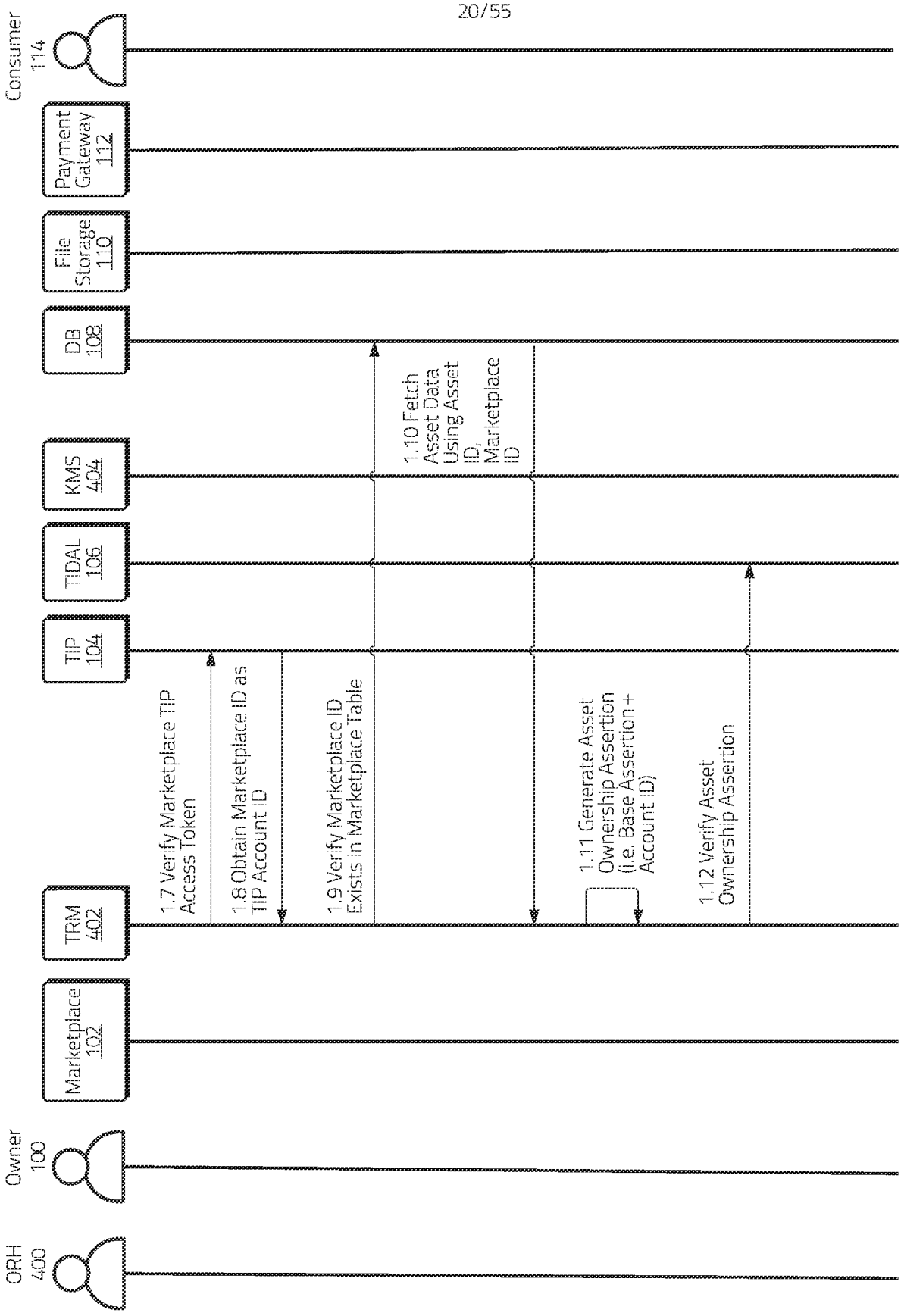

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
1.7 Verify Marketplace TIP Access Token
1.8 Obtain Marketplace ID as TIP Account ID
1.9 Verify Marketplace ID Exists in Marketplace Table
1.10 Fetch Asset Data Using Asset ID, Marketplace ID
1.11 Generate Asset Ownership Assertion (i.e. Base Assertion + Account ID)
1.12 Verify Asset Ownership Assertion

Figure 8B

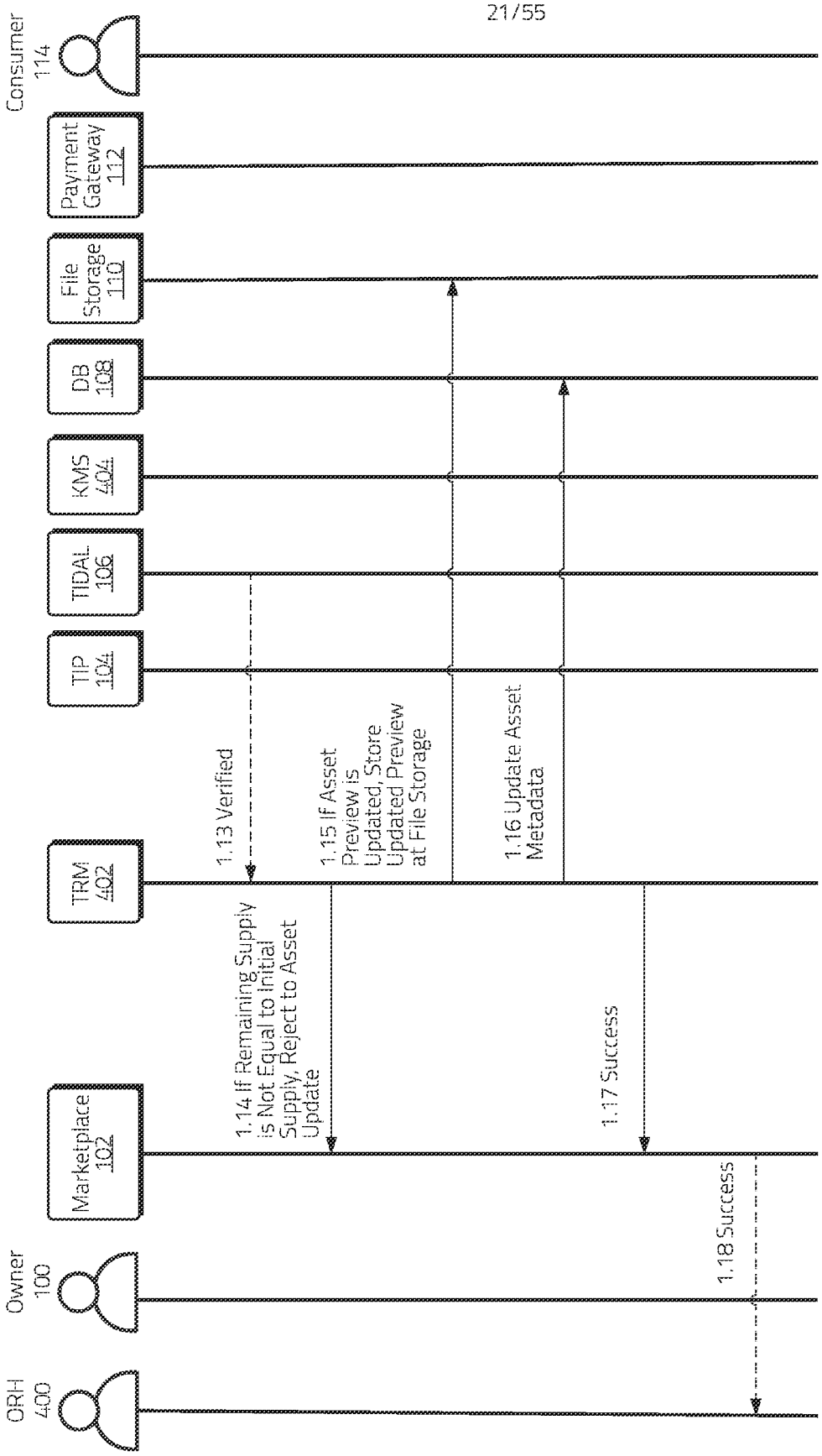

Consumer 114
Payment Gateway 112
File Storage 110
DB 108
KMS 404
TIDAL 106
TIP 104
TRM 402
Marketplace 102
Owner 100
ORH 400
1.13 Verified
1.14 If Remaining Supply is Not Equal to Initial Supply, Reject to Asset Update
1.15 If Asset Preview is Updated, Store Updated Preview at File Storage
1.16 Update Asset Metadata
1.17 Success
1.18 Success

Figure 8C

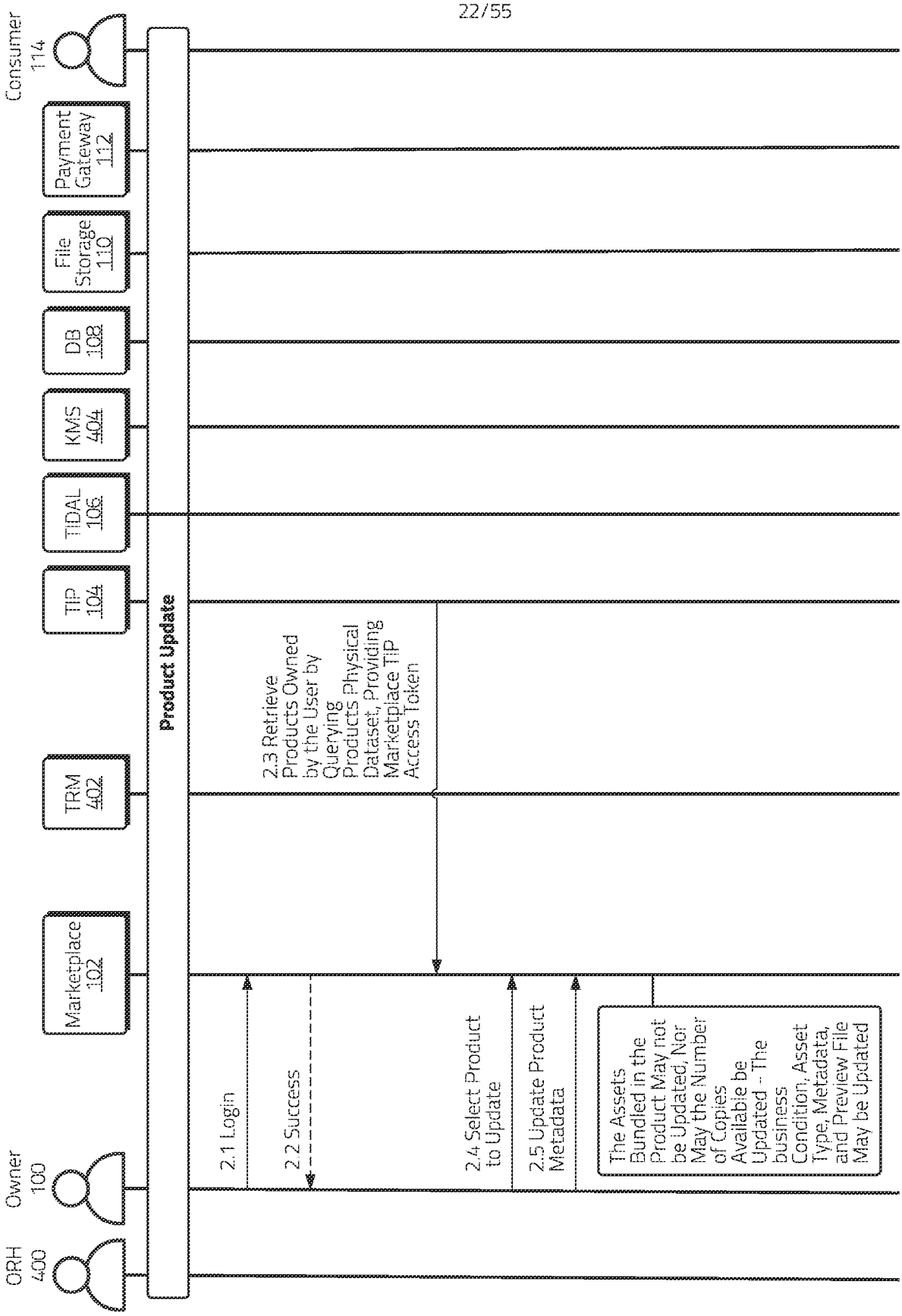

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Product Update
2.1 Login
2.2 Success
2.3 Retrieve Products Owned by the User by Querying Products Physical Dataset, Providing Marketplace TIP Access Token
2.4 Select Product to Update
2.5 Update Product Metadata
The Assets Bundled in the Product May not be Updated, Nor May the Number of Copies Available be Updated - The business Condition, Asset Type, Metadata, and Preview File May be Updated

Figure 9A

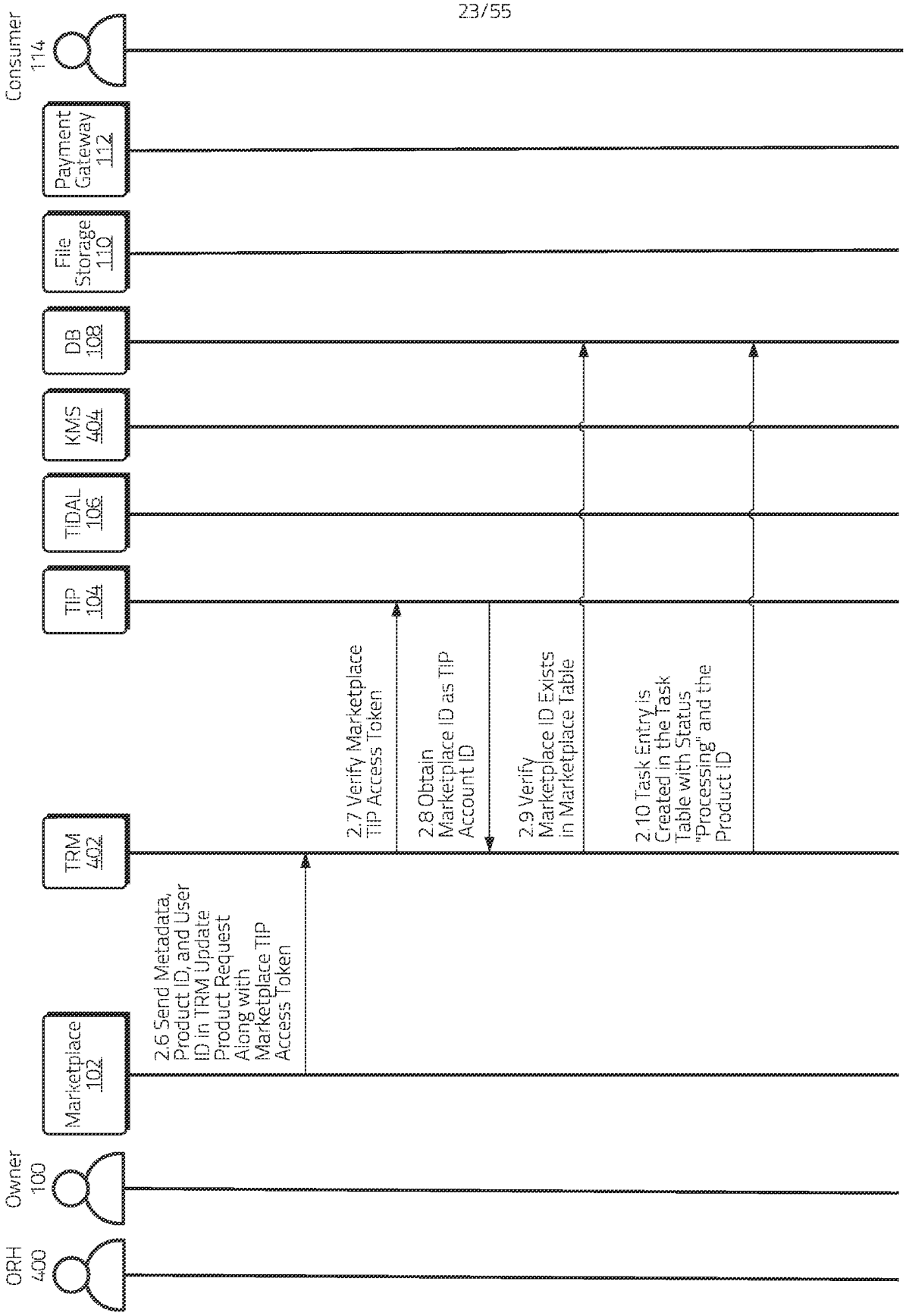

ORH 400
Owner 100
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
2.6 Send Metadata, Product ID, and User ID in TRM Update Product Request Along with Marketplace TIP Access Token
2.7 Verify Marketplace TIP Access Token
2.8 Obtain Marketplace ID as TIP Account ID
2.9 Verify Marketplace ID Exists in Marketplace Table
2.10 Task Entry is Created in the Task Table with Status "Processing" and the Product ID

Figure 9B

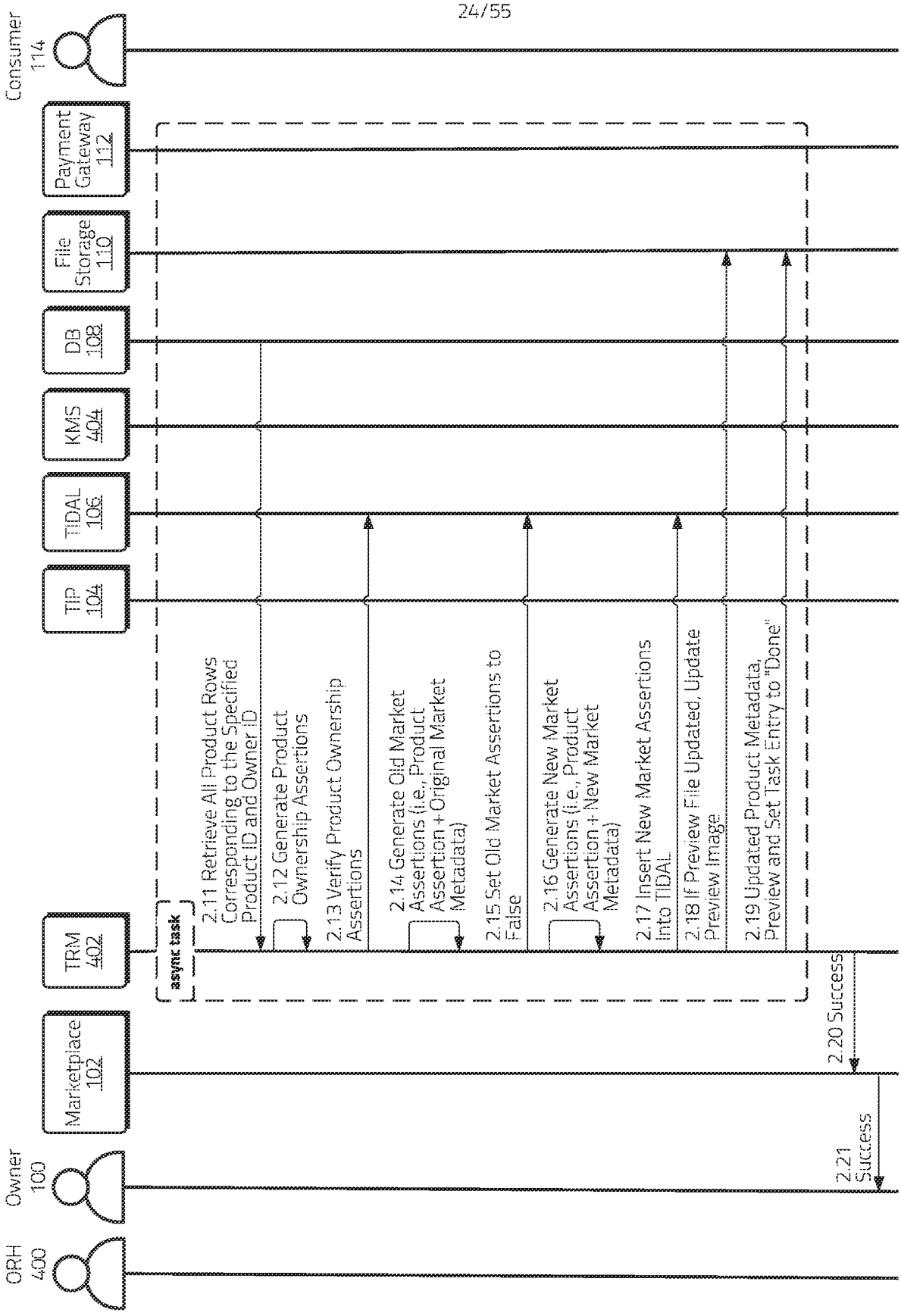

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
async task
2.11 Retrieve All Product Rows Corresponding to the Specified Product ID and Owner ID
2.12 Generate Product Ownership Assertions
2.13 Verify Product Ownership Assertions
2.14 Generate Old Market Assertions (i.e., Product Assertion + Original Market Metadata)
2.15 Set Old Market Assertions to False
2.16 Generate New Market Assertions (i.e., Product Assertion + New Market Metadata)
2.17 Insert New Market Assertions into TIDAL
2.18 If Preview File Updated, Update Preview Image
2.19 Updated Product Metadata, Preview and Set Task Entry to "Done"
2.20 Success
2.21 Success

Figure 9C

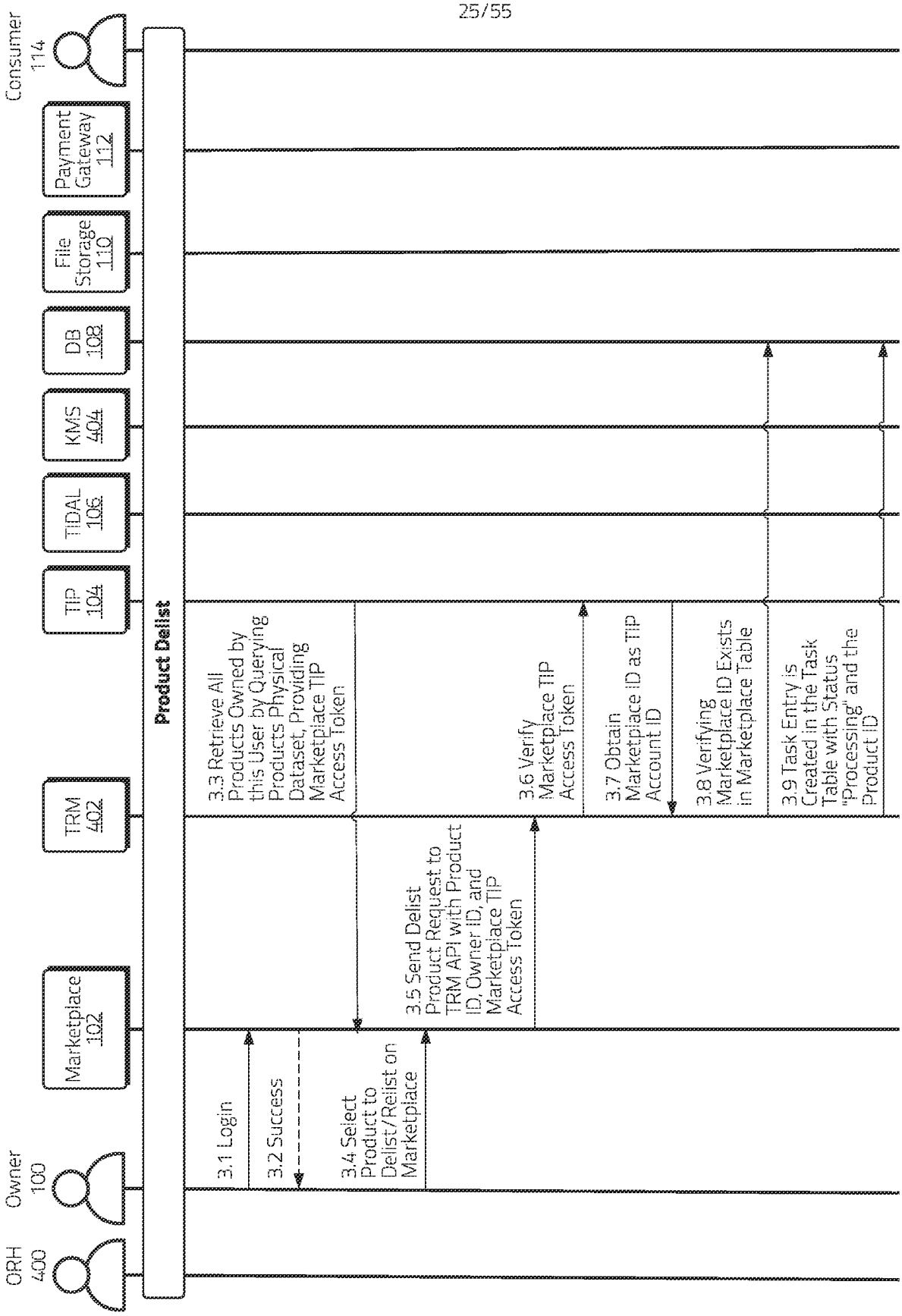

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Product Delist
3.1 Login
3.2 Success
3.4 Select Product to Delist/Relist on Marketplace
3.5 Send Delist Product Request to TRM API with Product ID, Owner ID, and Marketplace TIP Access Token
3.3 Retrieve All Products Owned by this User by Querying Products Physical Dataset, Providing Marketplace TIP Access Token
3.6 Verify Marketplace TIP Access Token
3.7 Obtain Marketplace ID as TIP Account ID
3.8 Verifying Marketplace ID Exists in Marketplace Table
3.9 Task Entry is Created in the Task Table with Status "Processing" and the Product ID

Figure 10A

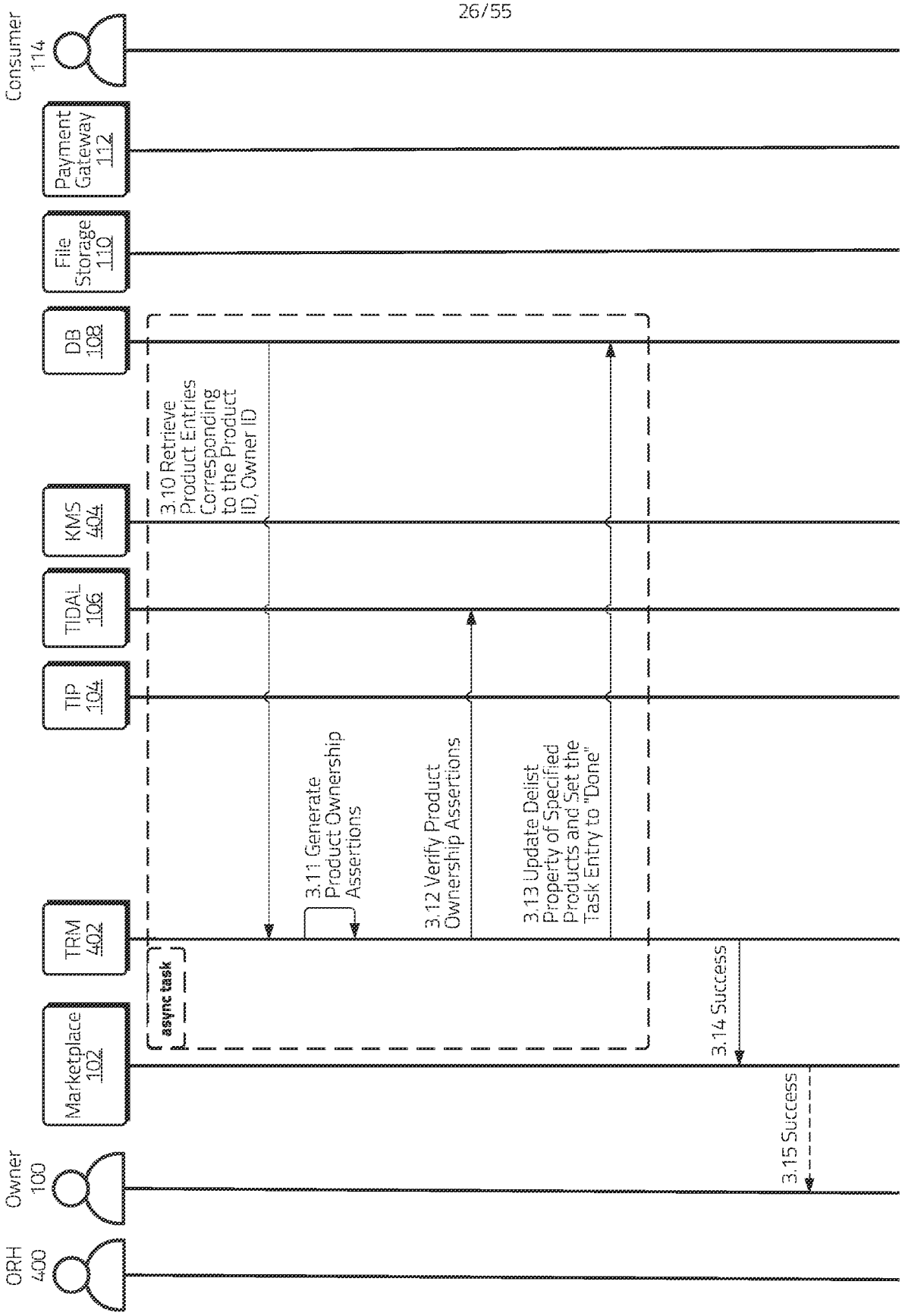

ORH 400
Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
async task
3.10 Retrieve Product Entries Corresponding to the Product ID, Owner ID
3.11 Generate Product Ownership Assertions
3.12 Verify Product Ownership Assertions
3.13 Update Delist Property of Specified Products and Set the Task Entry to "Done"
3.14 Success
3.15 Success

Figure 10B

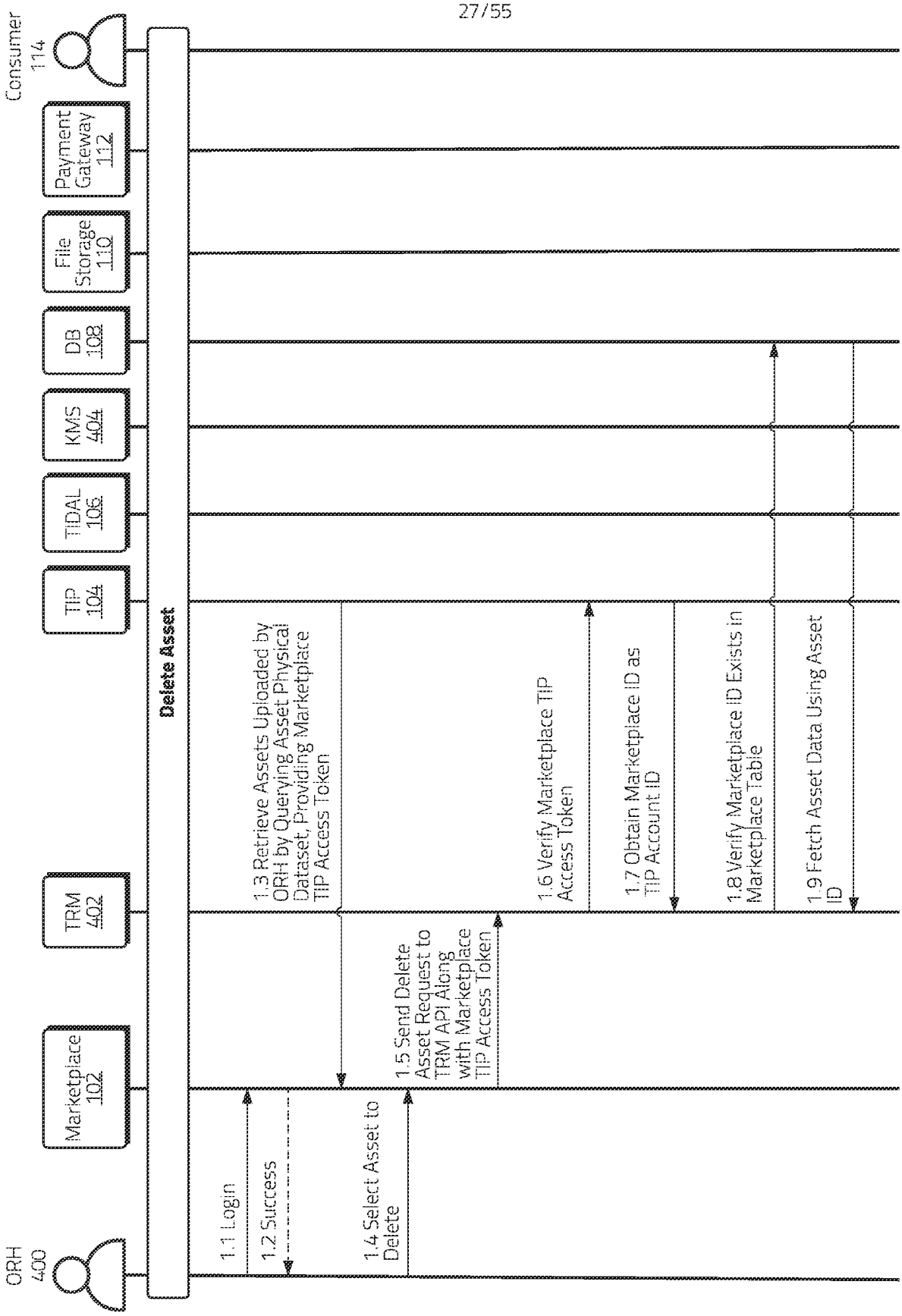

ORH 400
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Delete Asset
1.1 Login
1.2 Success
1.3 Retrieve Assets Uploaded by ORH by Querying Asset Physical Dataset, Providing Marketplace TIP Access Token
1.4 Select Asset to Delete
1.5 Send Delete Asset Request to TRM API Along with Marketplace TIP Access Token
1.6 Verify Marketplace TIP Access Token
1.7 Obtain Marketplace ID as TIP Account ID
1.8 Verify Marketplace ID Exists in Marketplace Table
1.9 Fetch Asset Data Using Asset ID

Figure 11A

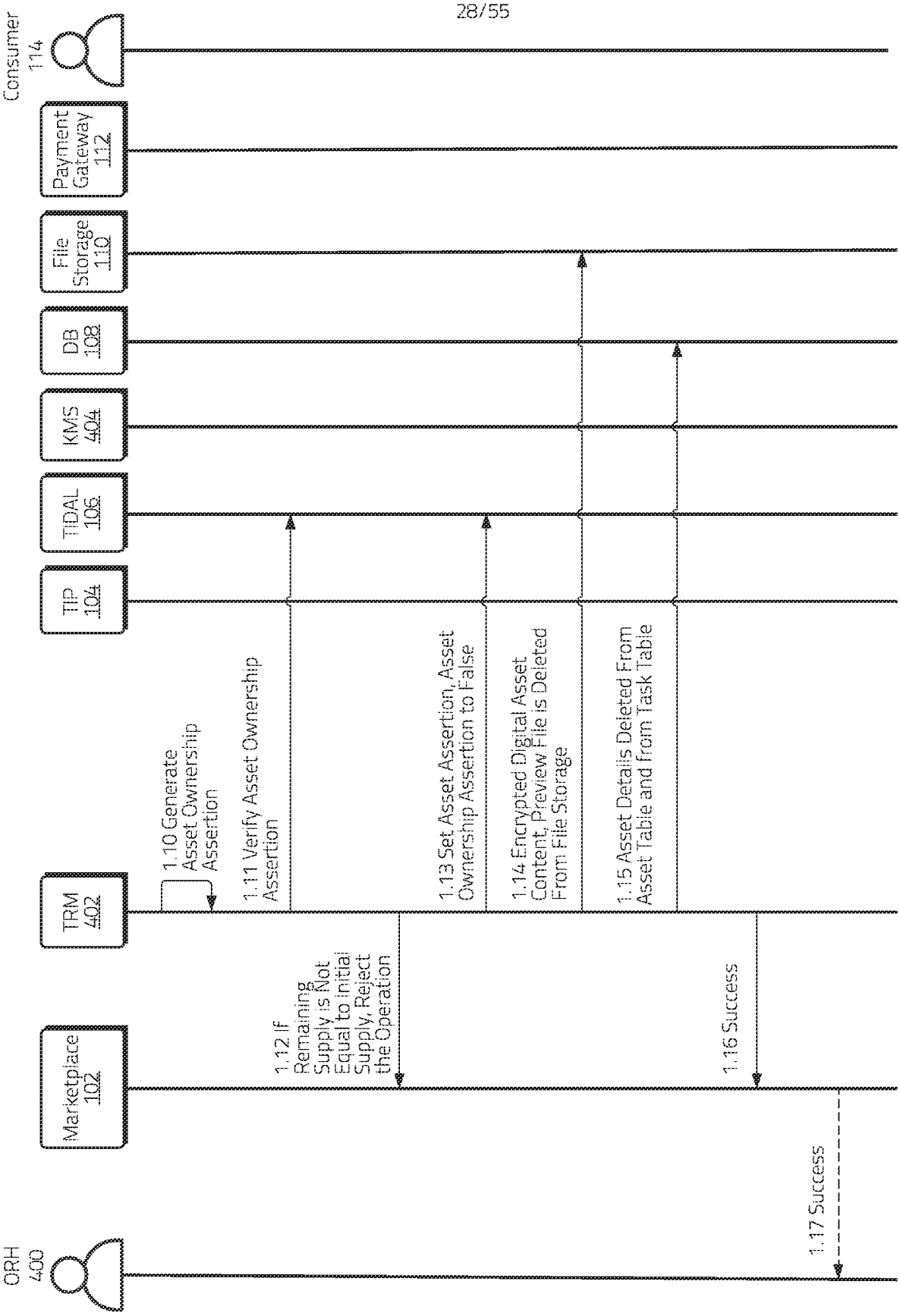

ORH 400
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
1.10 Generate Asset Ownership Assertion
1.11 Verify Asset Ownership Assertion
1.12 If Remaining Supply is Not Equal to initial Supply, Reject the Operation
1.13 Set Asset Assertion, Asset Ownership Assertion to False
1.14 Encrypted Digital Asset Content, Preview File is Deleted From File Storage
1.15 Asset Details Deleted From Asset Table and from Task Table
1.16 Success
1.17 Success

Figure 11B

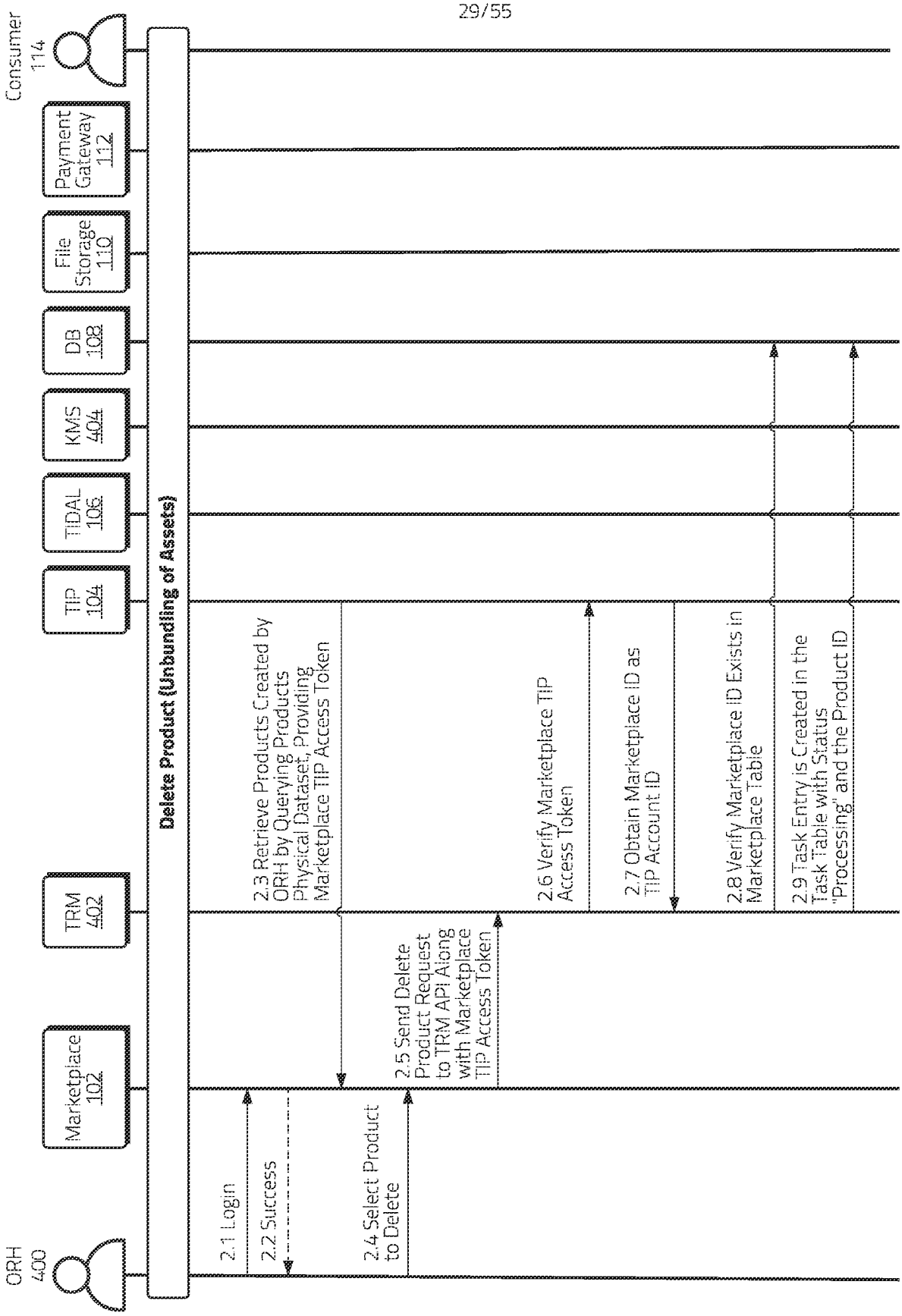

ORH 400
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Delete Product (Unbundling of Assets)
2.1 Login
2.2 Success
2.3 Retrieve Products Created by ORH by Querying Products Physical Dataset, Providing Marketplace TIP Access Token
2.4 Select Product to Delete
2.5 Send Delete Product Request to TRM API Along with Marketplace TIP Access Token
2.6 Verify Marketplace TIP Access Token
2.7 Obtain Marketplace ID as TIP Account ID
2.8 Verify Marketplace ID Exists in Marketplace Table
2.9 Task Entry is Created in the Task Table with Status "Processing" and the Product ID

Figure 12A

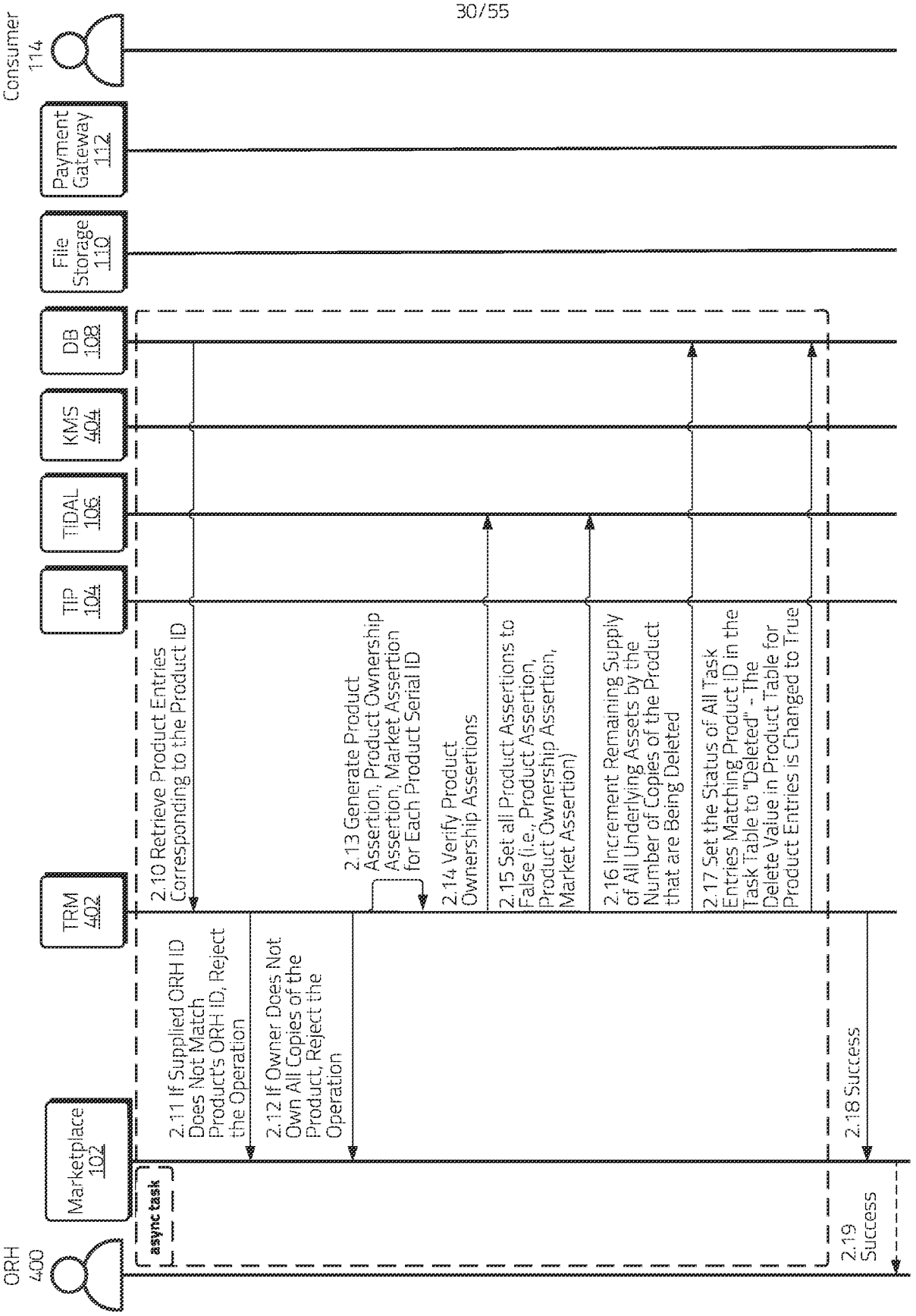

ORH 400
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
Consumer 114
async task
2.10 Retrieve Product Entries Corresponding to the Product ID
2.11 If Supplied ORH ID Does Not Match Product's ORH ID, Reject the Operation
2.12 If Owner Does Not Own All Copies of the Product, Reject the Operation
2.13 Generate Product Assertion, Product Ownership Assertion, Market Assertion for Each Product Serial ID
2.14 Verify Product Ownership Assertions
2.15 Set all Product Assertions to False (i.e., Product Assertion, Product Ownership Assertion, Market Assertion)
2.16 Increment Remaining Supply of All Underlying Assets by the Number of Copies of the Product that are Being Deleted
2.17 Set the Status of All Task Entries Matching Product ID in the Task Table to "Deleted" - The Delete Value in Product Table for Product Entries is Changed to True
2.18 Success
2.19 Success

Figure 12B

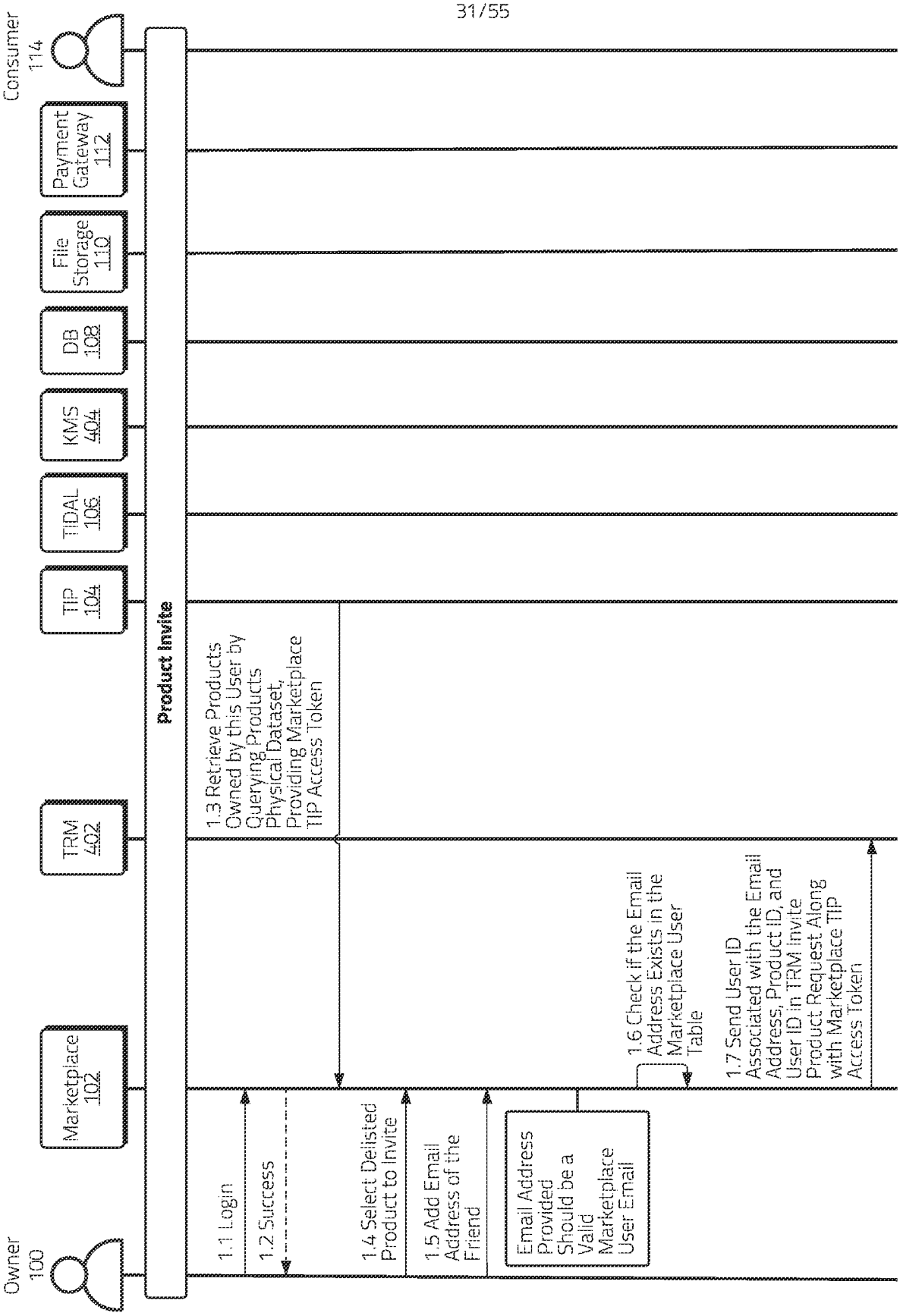

Owner
100
Consumer
114
Marketplace
102
TRM
402
TIP
104
TIDAL
106
KMS
404
DB
108
File
Storage
110
Payment
Gateway
112
Product Invite
1.1 Login
1.2 Success
1.3 Retrieve Products
Owned by this User by
Querying Products
Physical Dataset,
Providing Marketplace
TIP Access Token
1.4 Select Delisted
Product to Invite
1.5 Add Email
Address of the
Friend
Email Address
Provided
Should be a
Valid
Marketplace
User Email
1.6 Check if the Email
Address Exists in the
Marketplace User
Table
1.7 Send User ID
Associated with the Email
Address, Product ID, and
User ID in TRM invite
Product Request Along
with Marketplace TIP
Access Token

Figure 13A

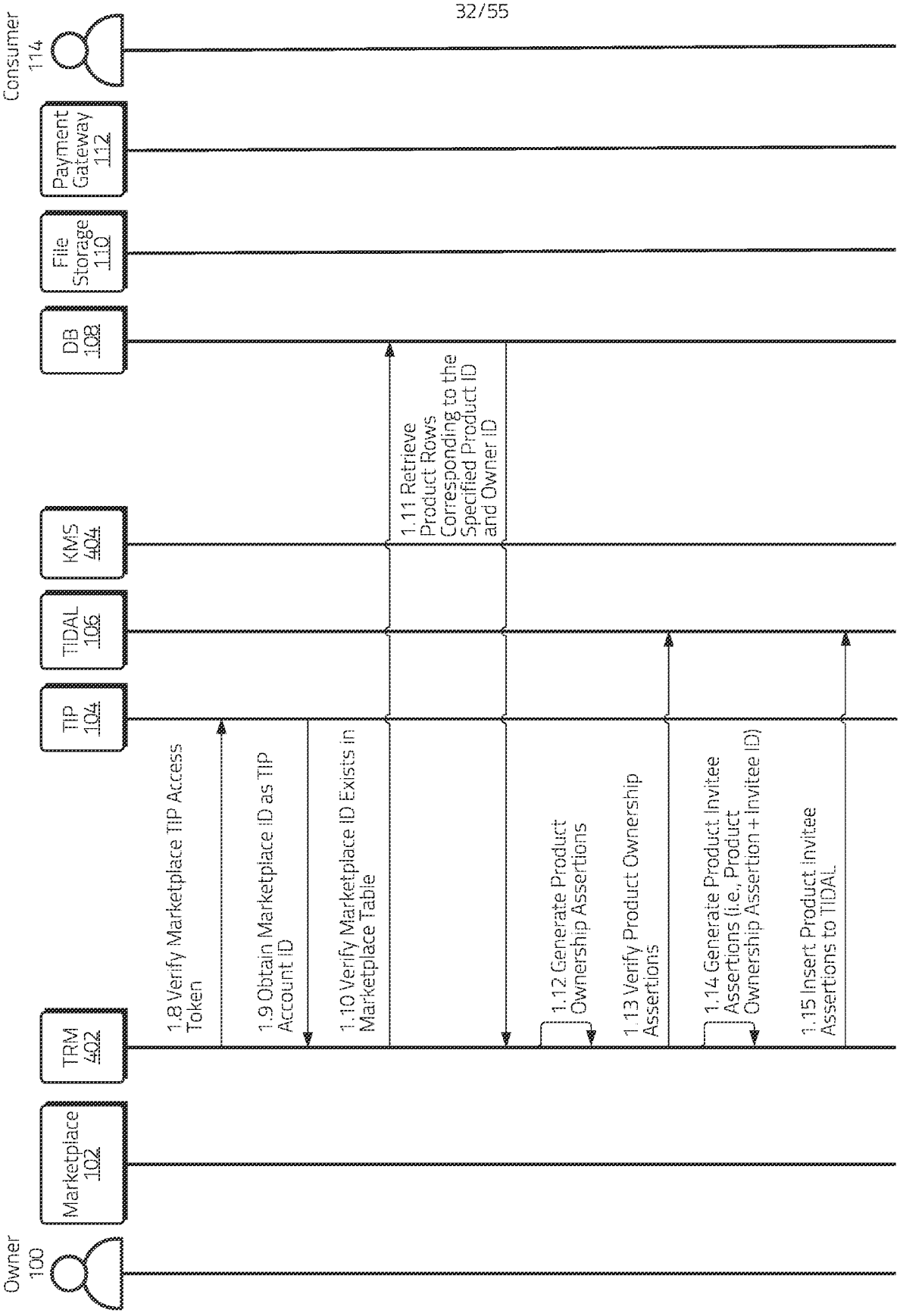

Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
1.8 Verify Marketplace TIP Access Token
1.9 Obtain Marketplace ID as TIP Account ID
1.10 Verify Marketplace ID Exists in Marketplace Table
1.11 Retrieve Product Rows Corresponding to the Specified Product ID and Owner ID
1.12 Generate Product Ownership Assertions
1.13 Verify Product Ownership Assertions
1.14 Generate Product Invitee Assertions (i.e., Product Ownership Assertion + Invitee ID)
1.15 Insert Product Invitee Assertions to TIDAL

Figure 13B

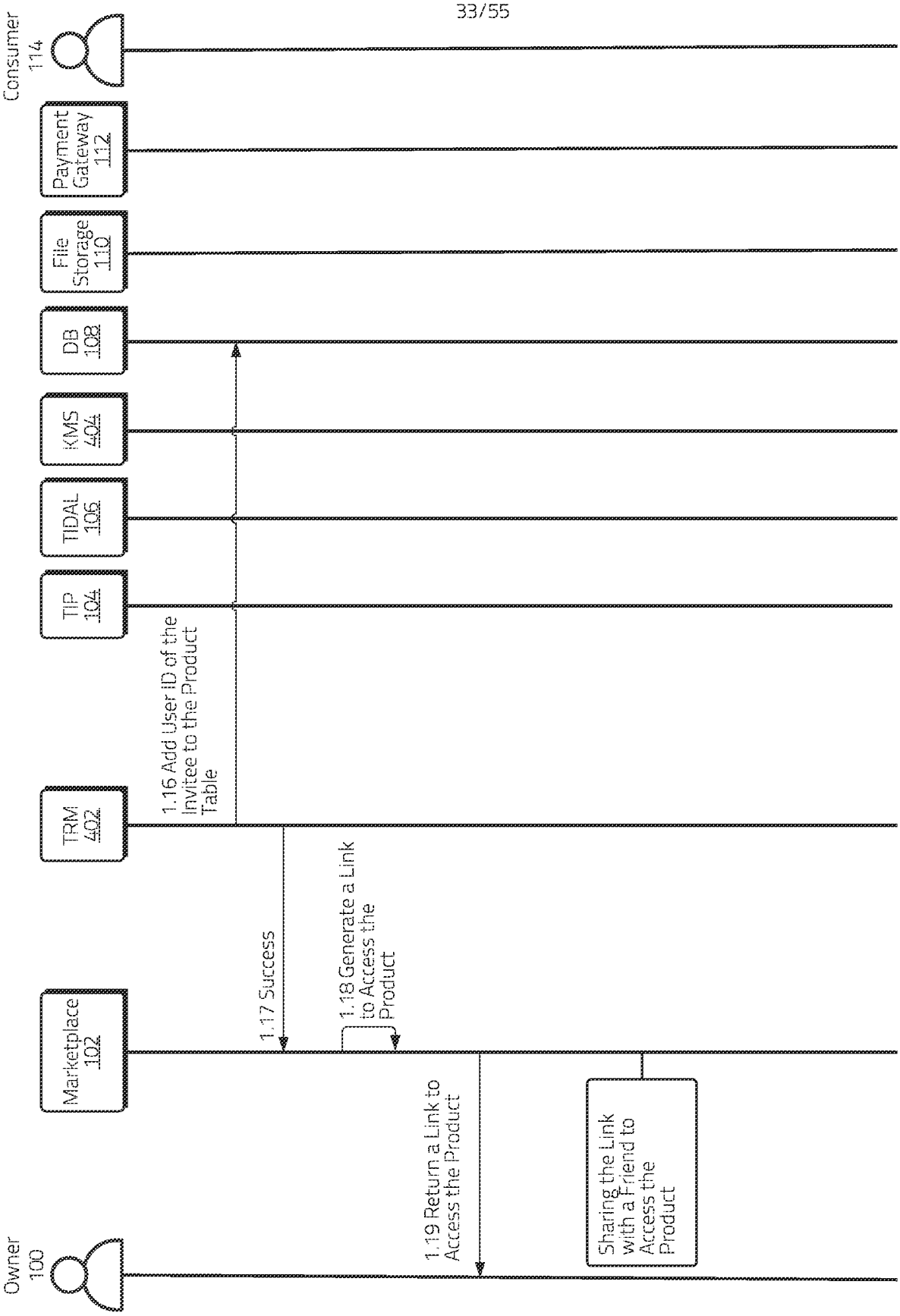

Consumer 114
Payment Gateway 112
File Storage 110
DB 108
KMS 404
TIDAL 106
TIP 104
TRM 402
Marketplace 102
Owner 100
1.16 Add User ID of the Invitee to the Product Table
1.17 Success
1.18 Generate a Link to Access the Product
1.19 Return a Link to Access the Product
Sharing the Link with a Friend to Access the Product

Figure 13C

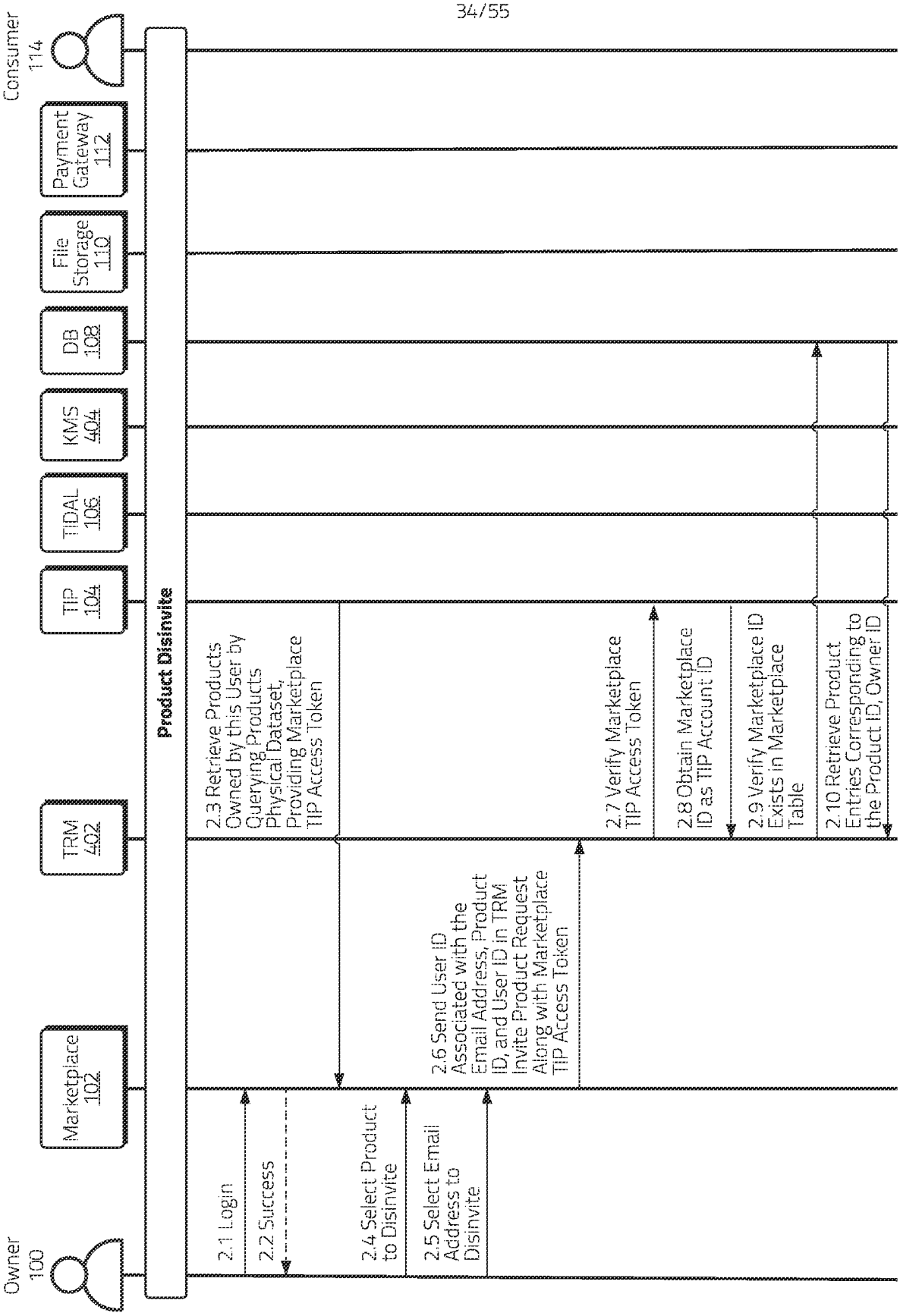

Owner
100
Consumer
114
Marketplace
102
TRM
402
TIP
104
TIDAL
106
KMS
404
DB
108
File
Storage
110
Payment
Gateway
112
Product Disinvite
2.1 Login
2.2 Success
2.3 Retrieve Products Owned by this User by Querying Products Physical Dataset, Providing Marketplace TIP Access Token
2.4 Select Product to Disinvite
2.5 Select Email Address to Disinvite
2.6 Send User ID Associated with the Email Address, Product ID, and User ID in TRM Invite Product Request Along with Marketplace TIP Access Token
2.7 Verify Marketplace TIP Access Token
2.8 Obtain Marketplace ID as TIP Account ID
2.9 Verify Marketplace ID Exists in Marketplace Table
2.10 Retrieve Product Entries Corresponding to the Product ID, Owner ID

Figure 14A

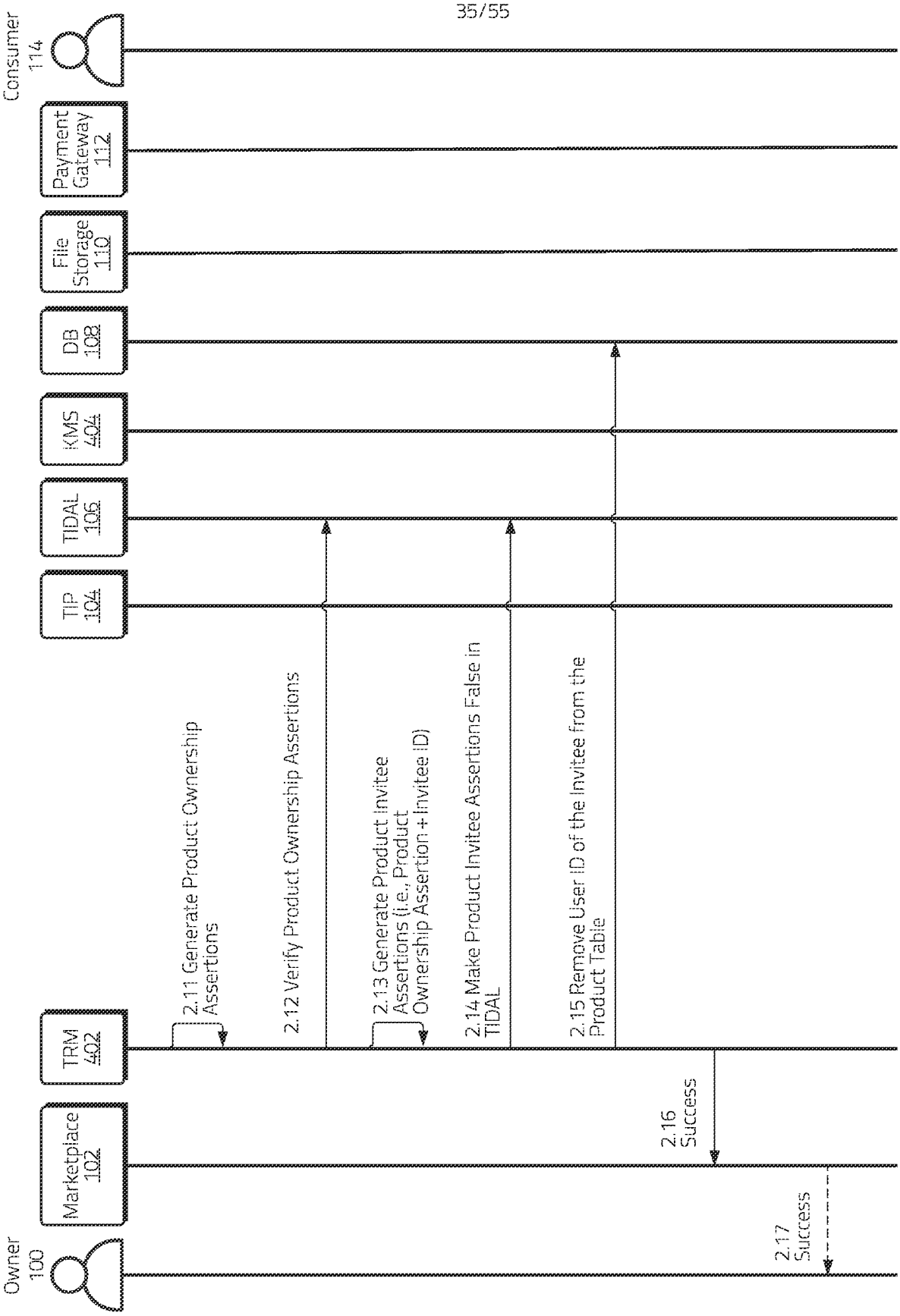

Owner 100
Consumer 114
Marketplace 102
TRM 402
TIP 104
TIDAL 106
KMS 404
DB 108
File Storage 110
Payment Gateway 112
2.11 Generate Product Ownership Assertions
2.12 Verify Product Ownership Assertions
2.13 Generate Product Invitee Assertions (i.e., Product Ownership Assertion + Invitee ID)
2.14 Make Product Invitee Assertions False in TIDAL
2.15 Remove User ID of the Invitee from the Product Table
2.16 Success
2.17 Success

Figure 14B

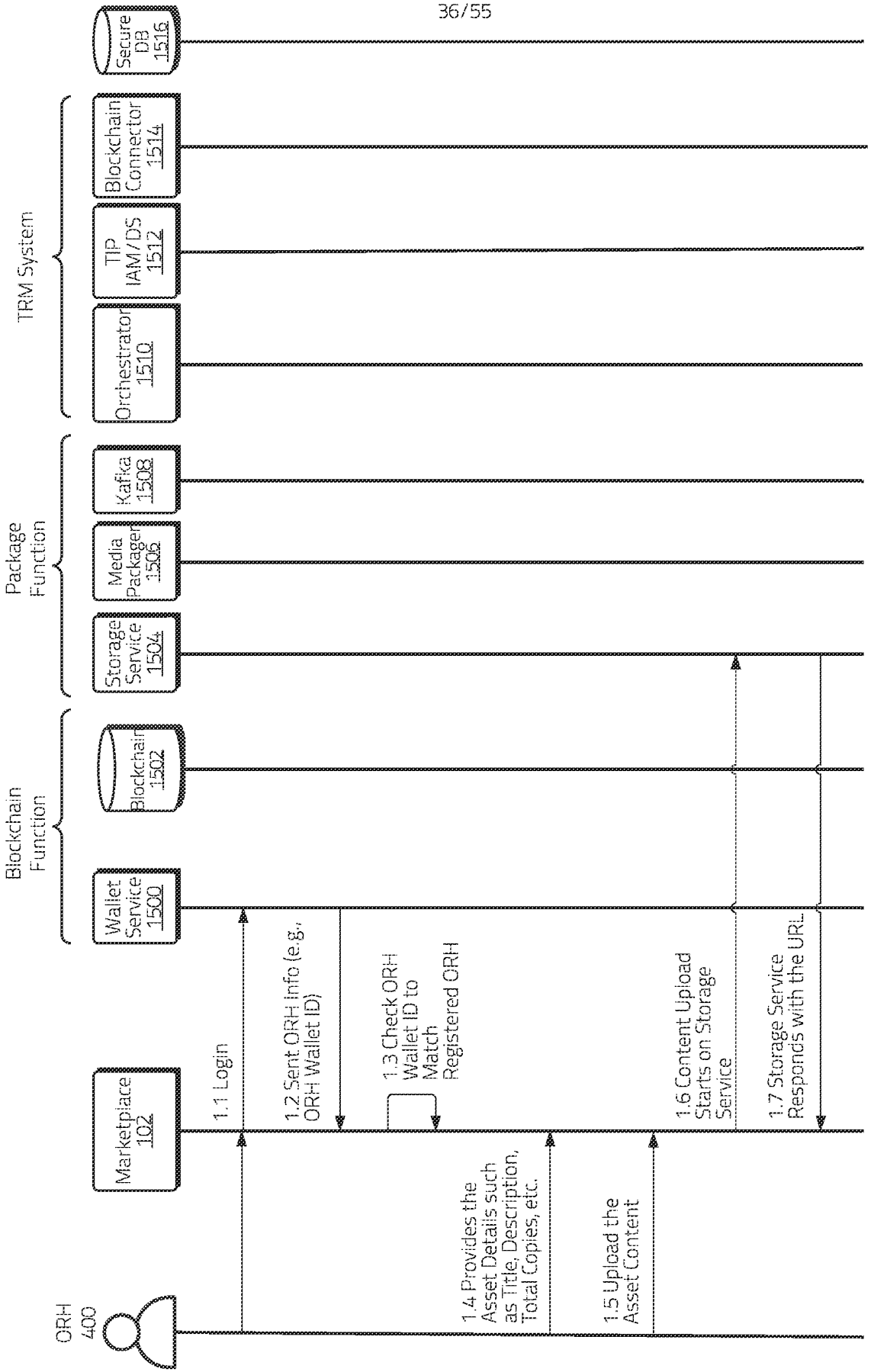

Blockchain Function
Package Function
TRM System
ORH 400
Marketplace 102
Wallet Service 1500
Blockchain 1502
Storage Service 1504
Media Packager 1506
Kafka 1508
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
1.1 Login
1.2 Sent ORH info (e.g., ORH Wallet ID)
1.3 Check ORH Wallet ID to Match Registered ORH
1.4 Provides the Asset Details such as Title, Description, Total Copies, etc.
1.5 Upload the Asset Content
1.6 Content Upload Starts on Storage Service
1.7 Storage Service Responds with the URL

Figure 15A

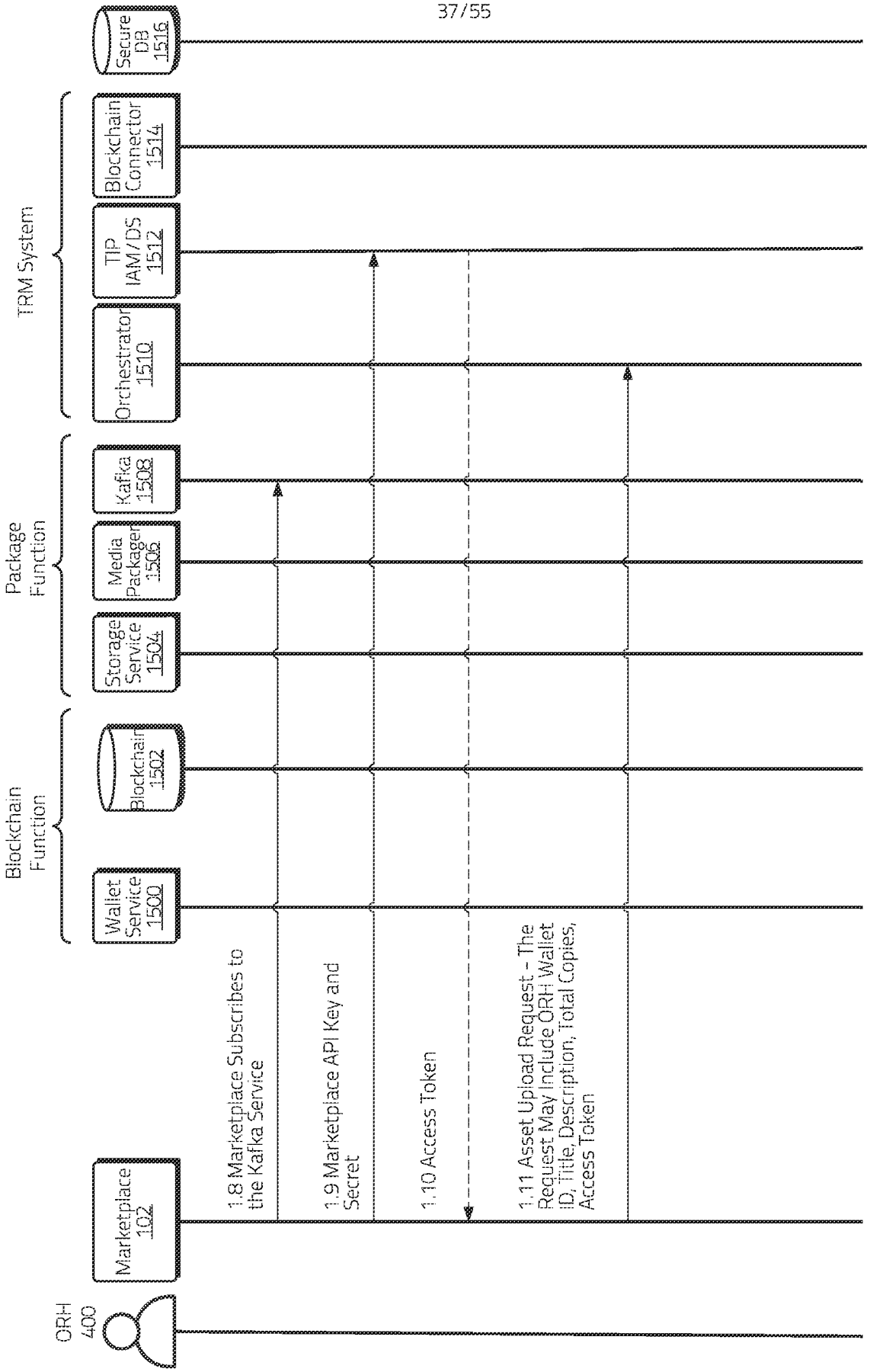

ORH 400
Marketplace 102
Wallet Service 1500
Blockchain 1502
Storage Service 1504
Media Packager 1506
Kafka 1508
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
Blockchain Function
Package Function
TRM System
1.8 Marketplace Subscribes to the Kafka Service
1.9 Marketplace API Key and Secret
1.10 Access Token
1.11 Asset Upload Request - The Request May include ORH Wallet ID, Title, Description, Total Copies, Access Token

Figure 15B

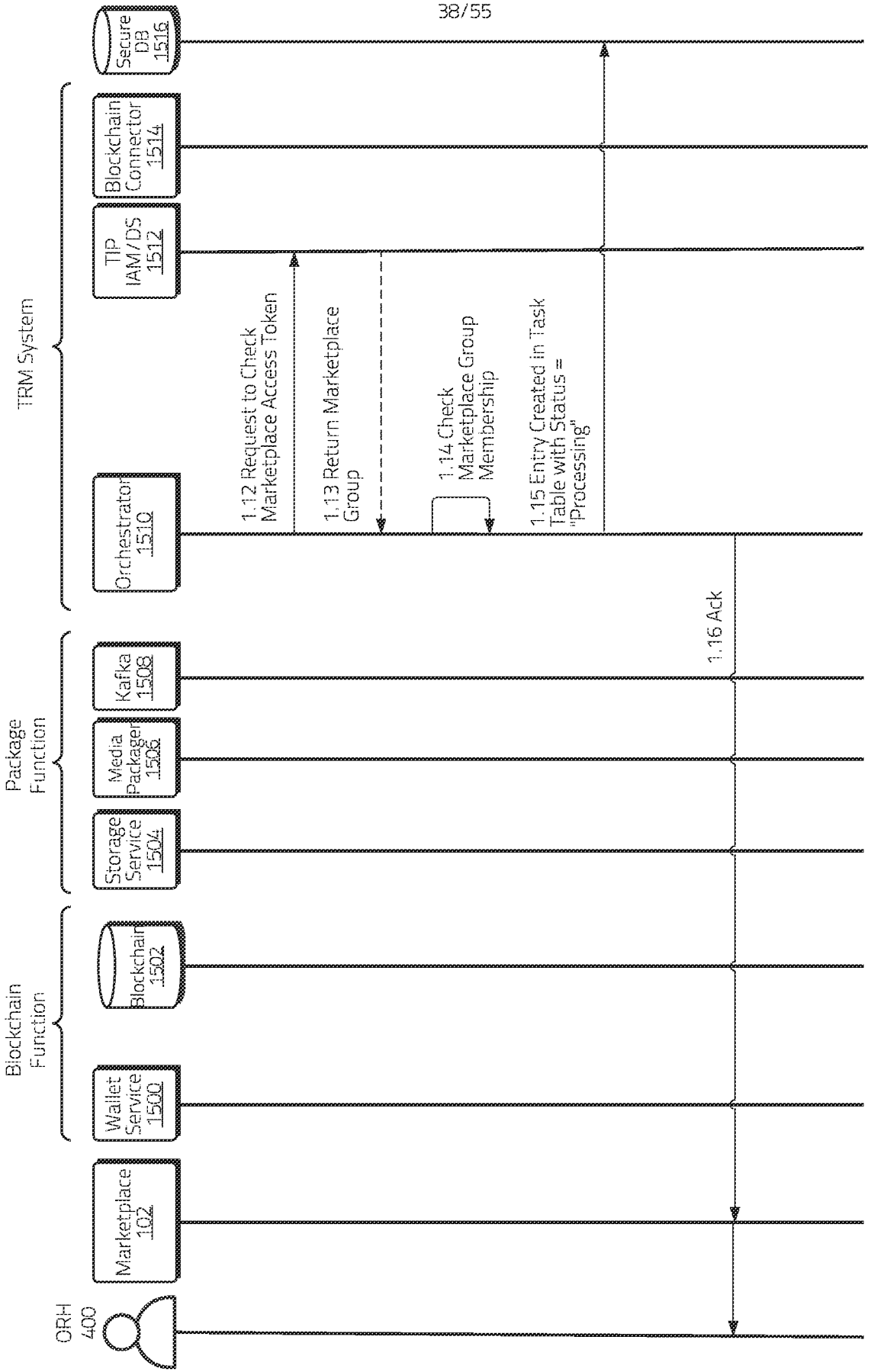

ORH 400
Marketplace 102
Blockchain Function
Wallet Service 1500
Blockchain 1502
Package Function
Storage Service 1504
Media Packager 1506
Kafka 1508
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
1.12 Request to Check Marketplace Access Token
1.13 Return Marketplace Group
1.14 Check Marketplace Group Membership
1.15 Entry Created in Task Table with Status = "Processing"
1.16 Ack

Figure 15C

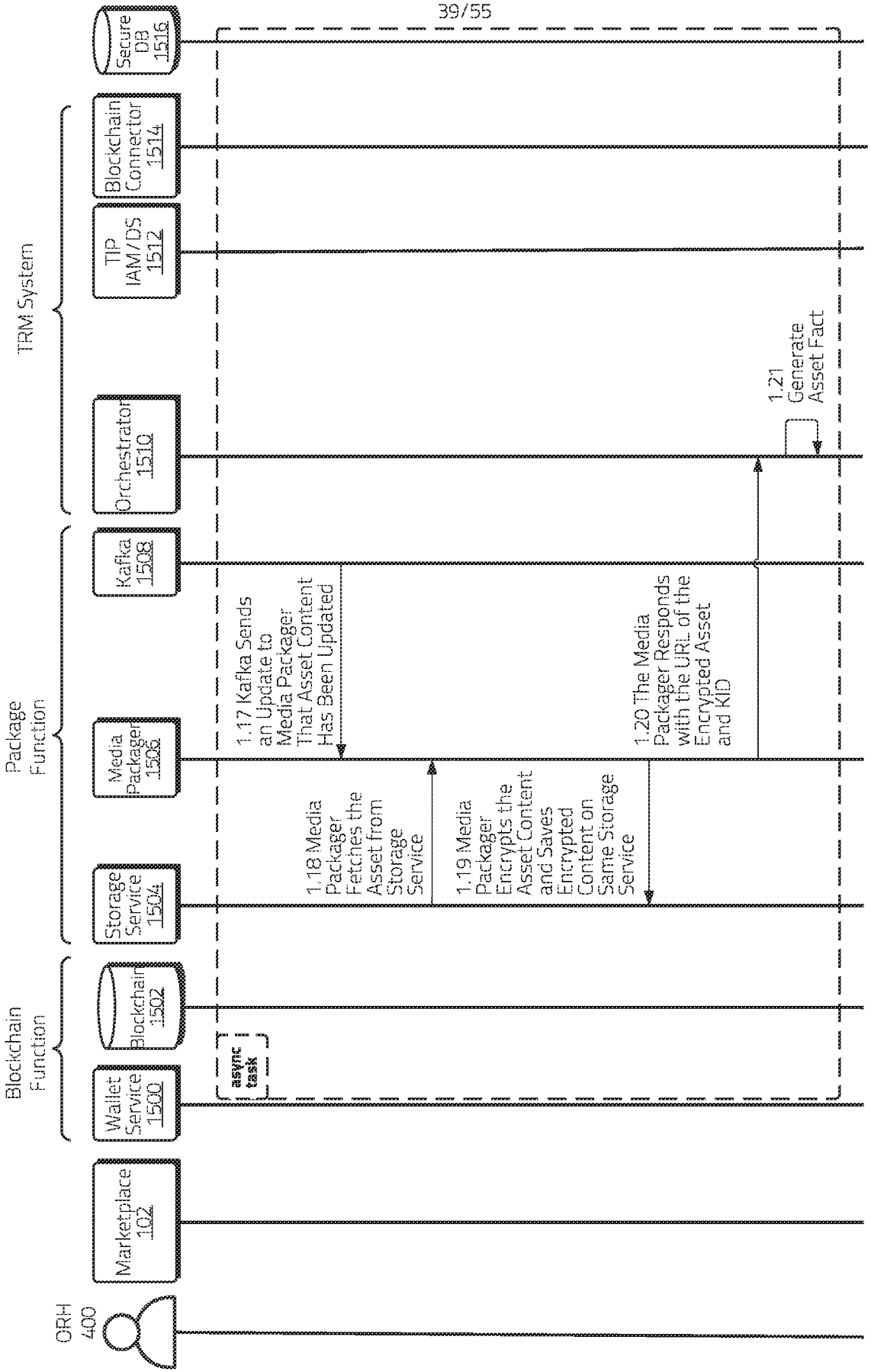

ORH 400
Marketplace 102
Blockchain Function
Wallet Service 1500
Blockchain 1502
async task
Package Function
Storage Service 1504
Media Packager 1506
Kafka 1508
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
1.17 Kafka Sends an Update to Media Packager That Asset Content Has Been Updated
1.18 Media Packager Fetches the Asset from Storage Service
1.19 Media Packager Encrypts the Asset Content and Saves Encrypted Content on Same Storage Service
1.20 The Media Packager Responds with the URL of the Encrypted Asset and KID
1.21 Generate Asset Fact

Figure 15D

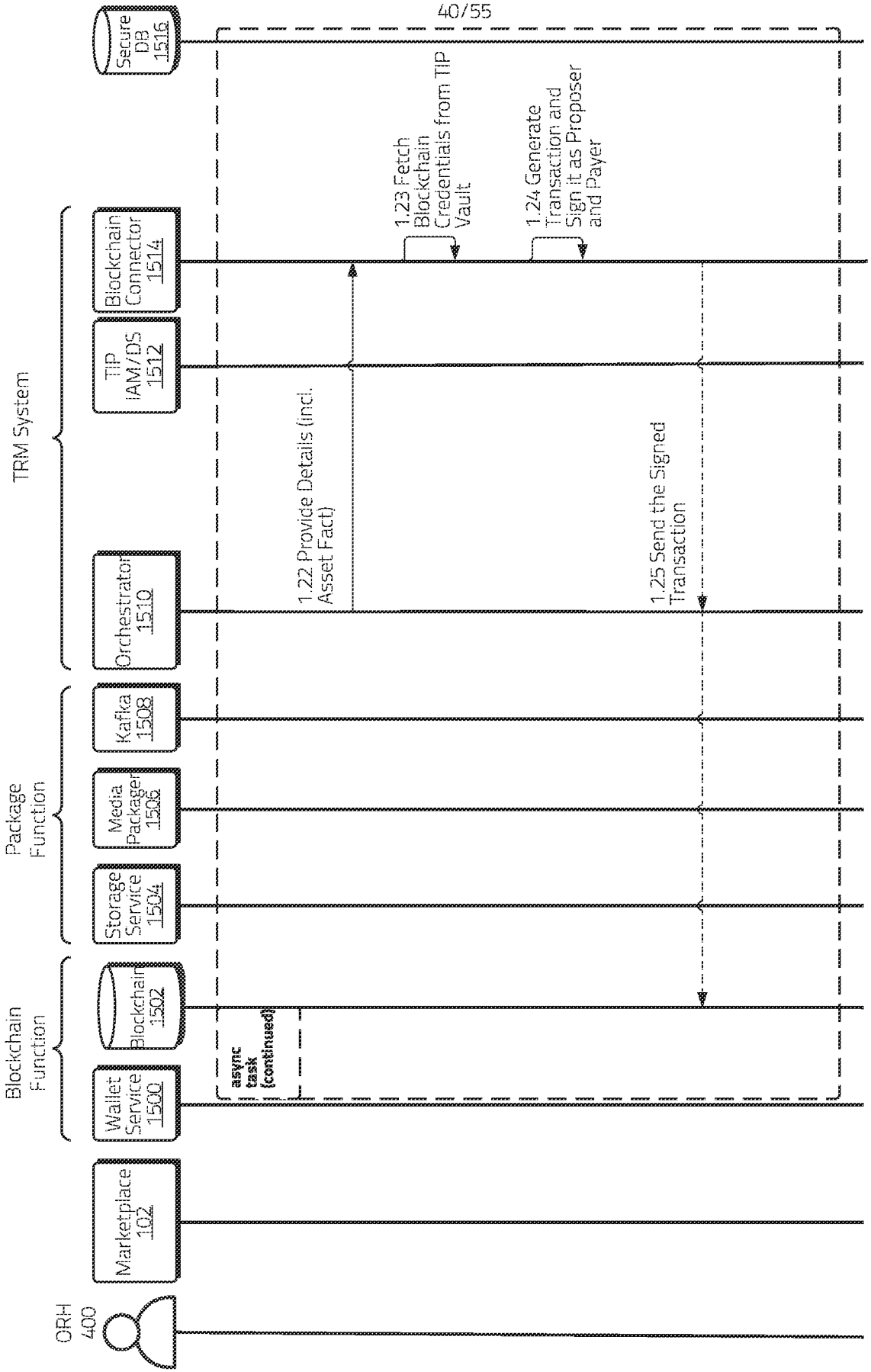

ORH 400
Marketplace 102
Blockchain Function
Wallet Service 1500
Blockchain 1502
Package Function
Storage Service 1504
Media Packager 1506
Kafka 1508
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
async task (continued)
1.22 Provide Details (incl. Asset Fact)
1.23 Fetch Blockchain Credentials from TIP Vault
1.24 Generate Transaction and Sign it as Proposer and Payer
1.25 Send the Signed Transaction

Figure 15E

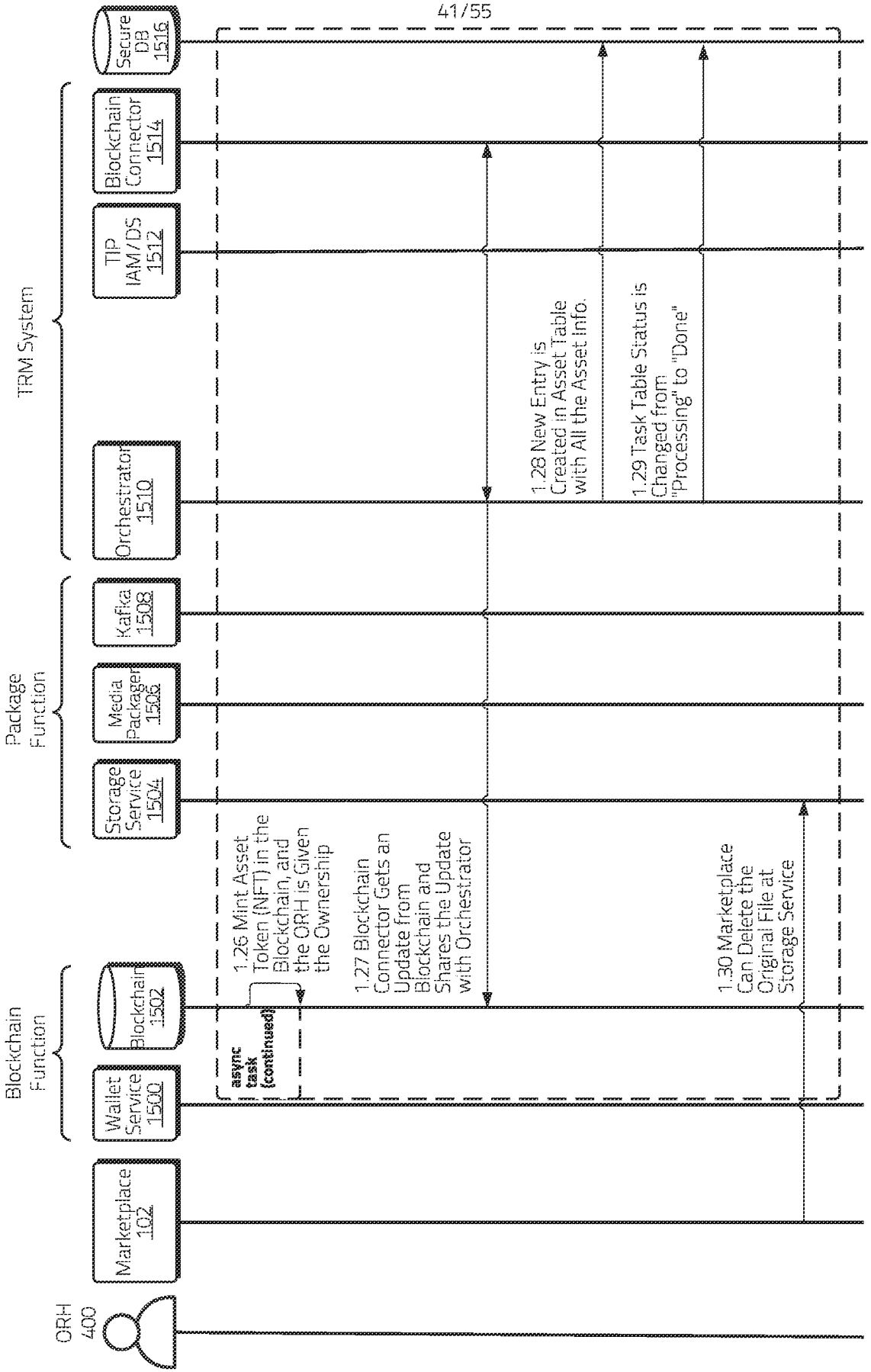

ORH 400
Marketplace 102
Blockchain Function
Wallet Service 1500
Blockchain 1502
Package Function
Storage Service 1504
Media Packager 1506
Kafka 1508
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
async task (continued)
1.26 Mint Asset Token (NFT) in the Blockchain, and the ORH is Given the Ownership
1.27 Blockchain Connector Gets an Update from Blockchain and Shares the Update with Orchestrator
1.28 New Entry is Created in Asset Table with All the Asset Info.
1.29 Task Table Status is Changed from "Processing" to "Done"
1.30 Marketplace Can Delete the Original File at Storage Service

Figure 15F

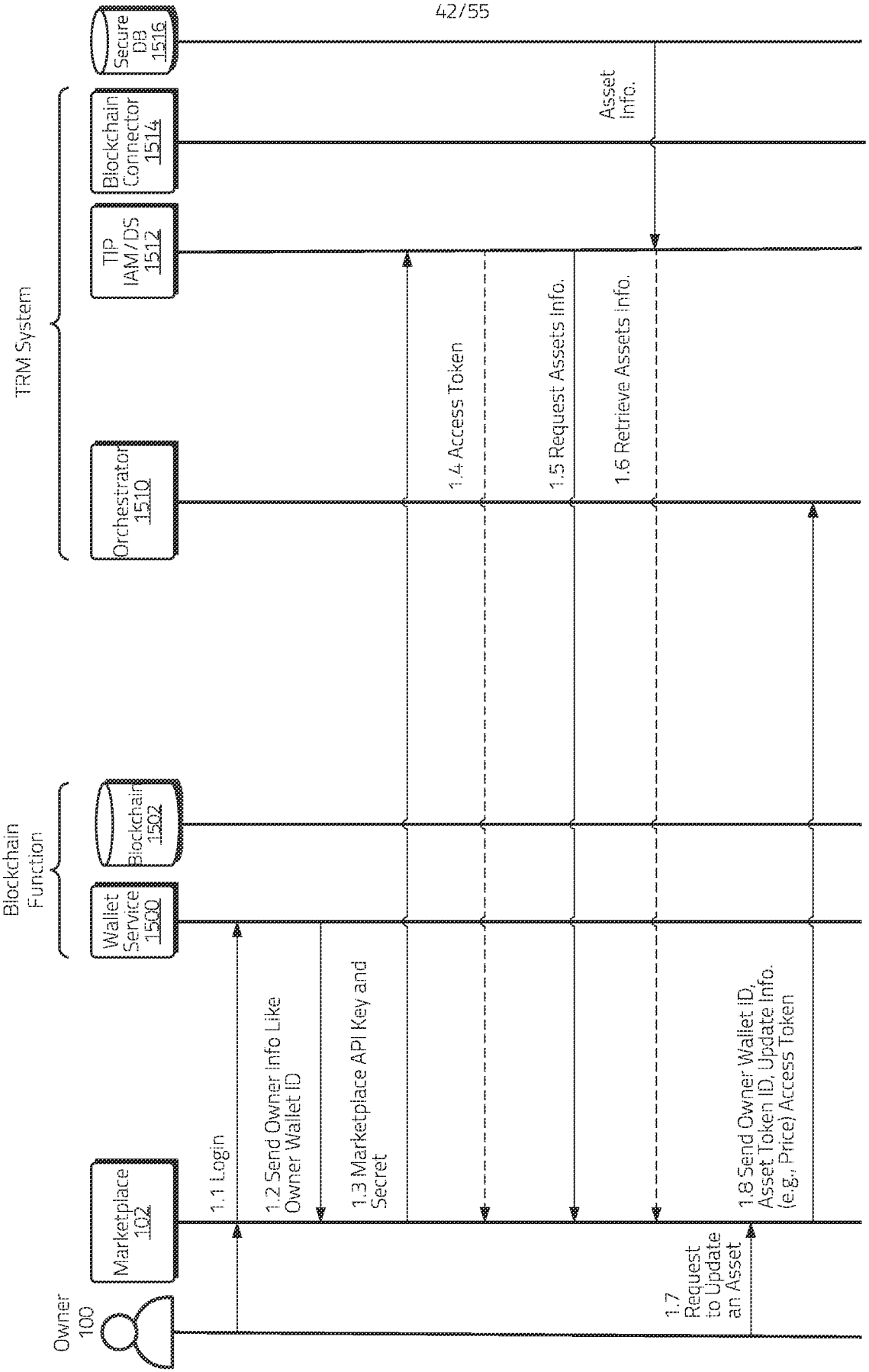

Owner 100
Marketplace 102
Blockchain Function
Wallet Service 1500
Blockchain 1502
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
1.1 Login
1.2 Send Owner Info Like Owner Wallet ID
1.3 Marketplace API Key and Secret
1.4 Access Token
1.5 Request Assets Info.
1.6 Retrieve Assets Info.
Asset Info.
1.7 Request to Update an Asset
1.8 Send Owner Wallet ID, Asset Token ID, Update Info. (e.g., Price) Access Token

Figure 16A

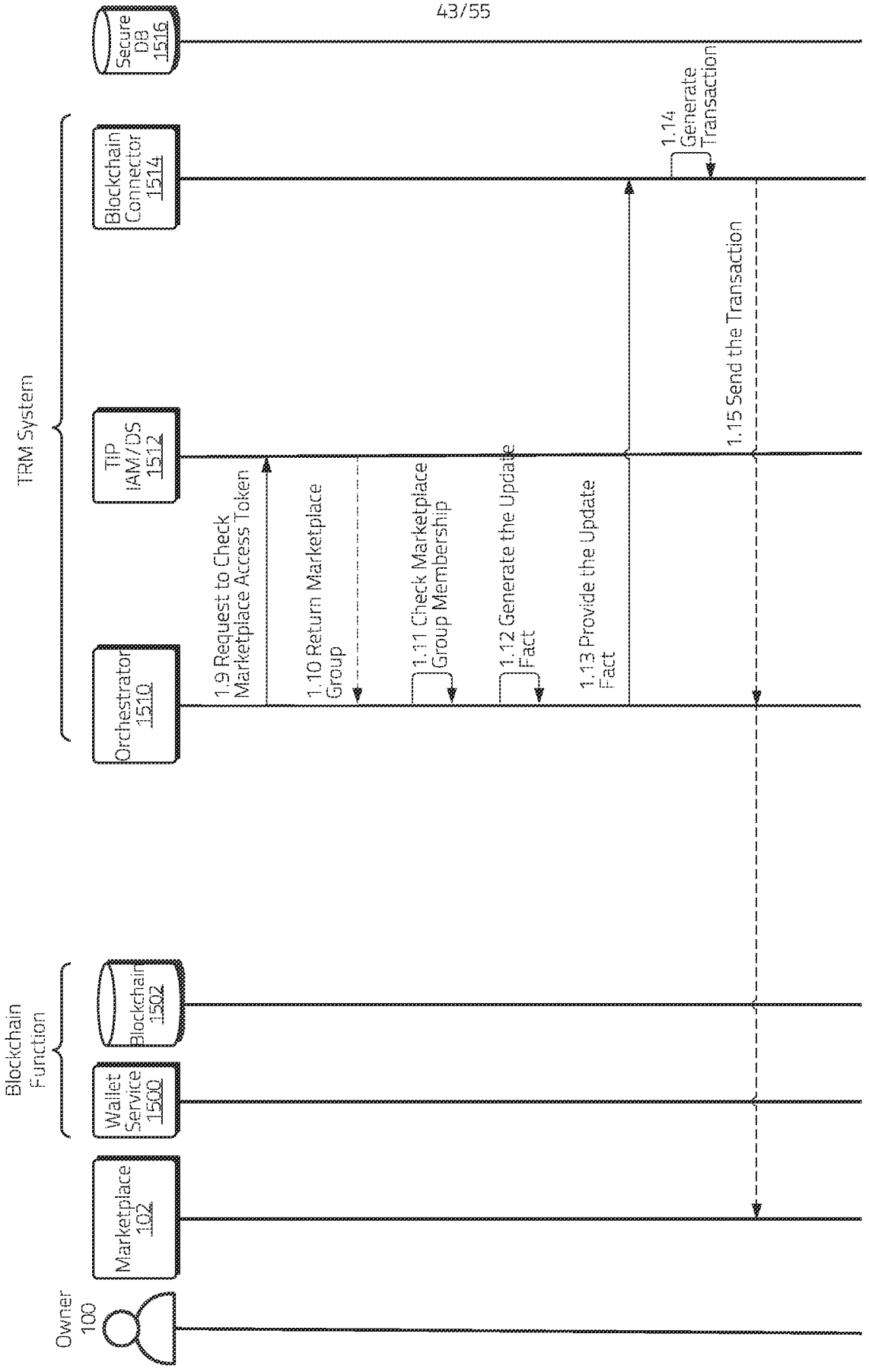

Owner 100
Blockchain Function
Marketplace 102
Wallet Service 1500
Blockchain 1502
TRM System
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
1.9 Request to Check Marketplace Access Token
1.10 Return Marketplace Group
1.11 Check Marketplace Group Membership
1.12 Generate the Update Fact
1.13 Provide the Update Fact
1.14 Generate Transaction
1.15 Send the Transaction

Figure 16B

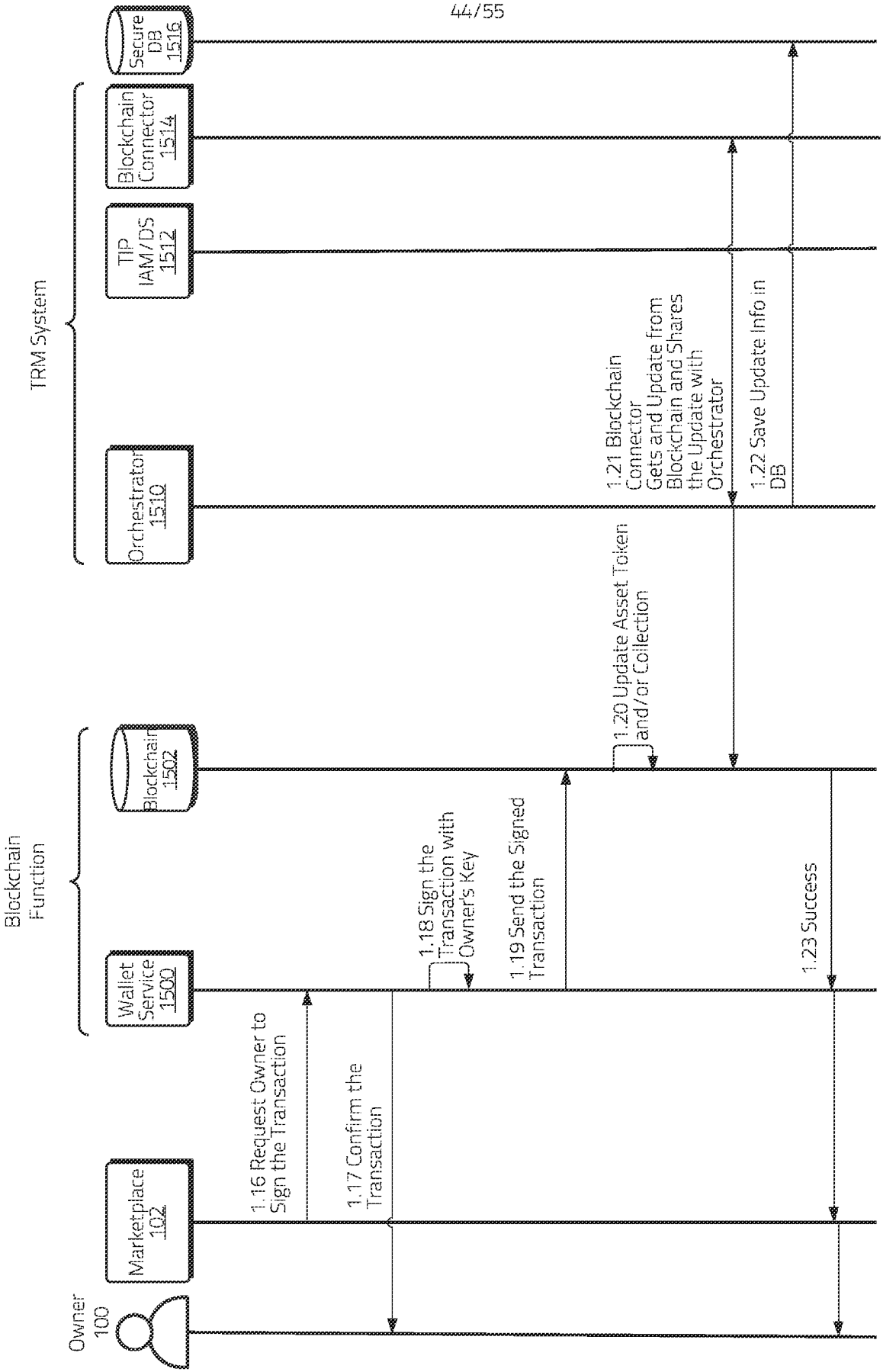

Owner
100
Marketplace
102
Blockchain
Function
Wallet
Service
1500
Blockchain
1502
TRM System
Orchestrator
1510
TIP
IAM/DS
1512
Blockchain
Connector
1514
Secure
DB
1516
1.16 Request Owner to Sign the Transaction
1.17 Confirm the Transaction
1.18 Sign the Transaction with Owner's Key
1.19 Send the Signed Transaction
1.20 Update Asset Token and/or Collection
1.21 Blockchain Connector Gets and Update from Blockchain and Shares the Update with Orchestrator
1.22 Save Update Info in DB
1.23 Success

Figure 16C

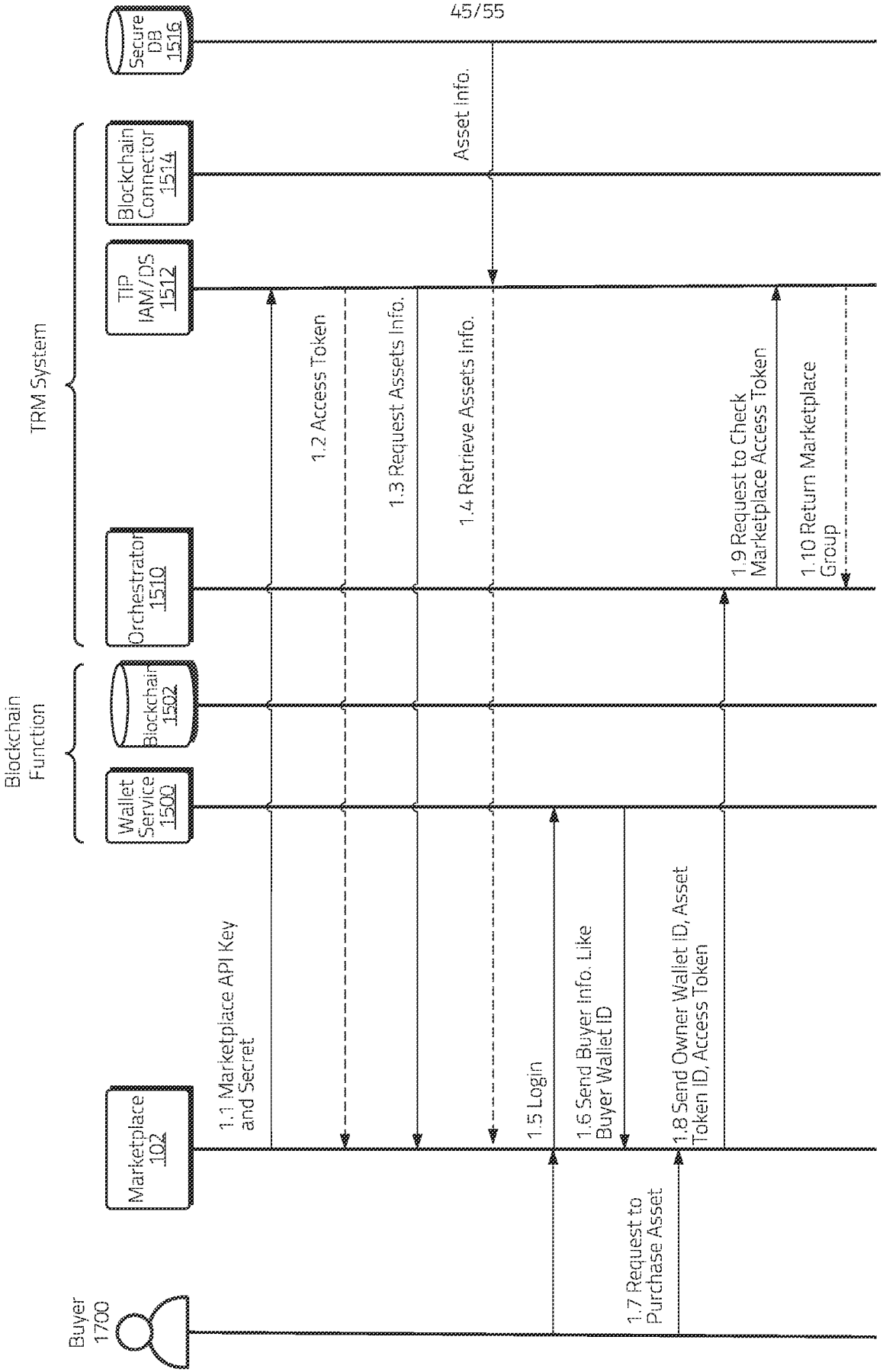

Buyer 1700
Marketplace 102
Wallet Service 1500
Blockchain 1502
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
Blockchain Function
TRM System
1.1 Marketplace API Key and Secret
1.2 Access Token
1.3 Request Assets Info.
1.4 Retrieve Assets Info.
1.5 Login
1.6 Send Buyer Info. Like Buyer Wallet ID
1.7 Request to Purchase Asset
1.8 Send Owner Wallet ID, Asset Token ID, Access Token
1.9 Request to Check Marketplace Access Token
1.10 Return Marketplace Group
Asset Info.

Figure 17A

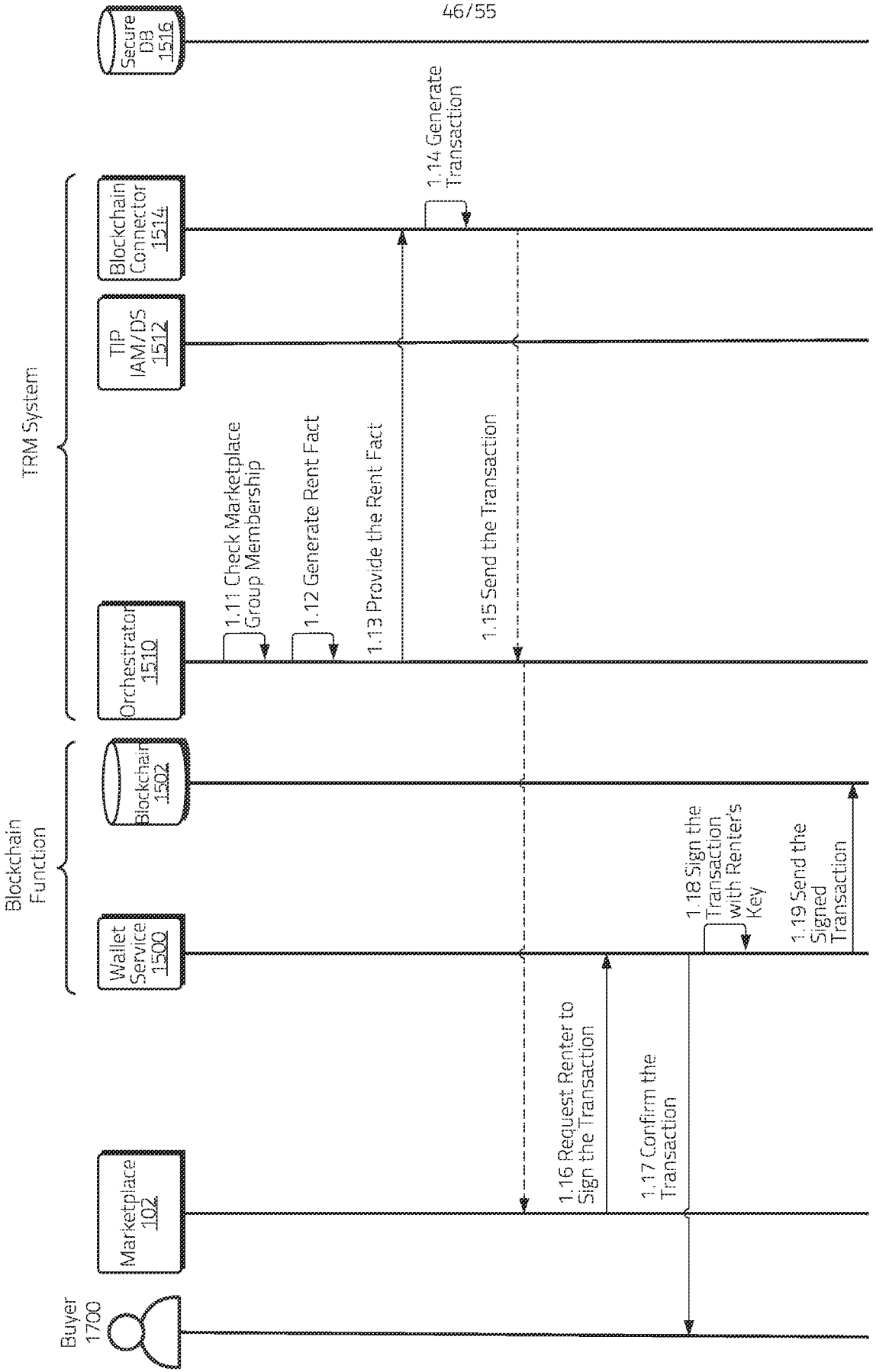

Buyer 1700
Marketplace 102
Wallet Service 1500
Blockchain 1502
Orchestrator 1510
Blockchain Connector 1514
TIP IAM/DS 1512
Secure DB 1516
Blockchain Function
TRM System
1.11 Check Marketplace Group Membership
1.12 Generate Rent Fact
1.13 Provide the Rent Fact
1.14 Generate Transaction
1.15 Send the Transaction
1.16 Request Renter to Sign the Transaction
1.17 Confirm the Transaction
1.18 Sign the Transaction with Renter's Key
1.19 Send the Signed Transaction

Figure 17B

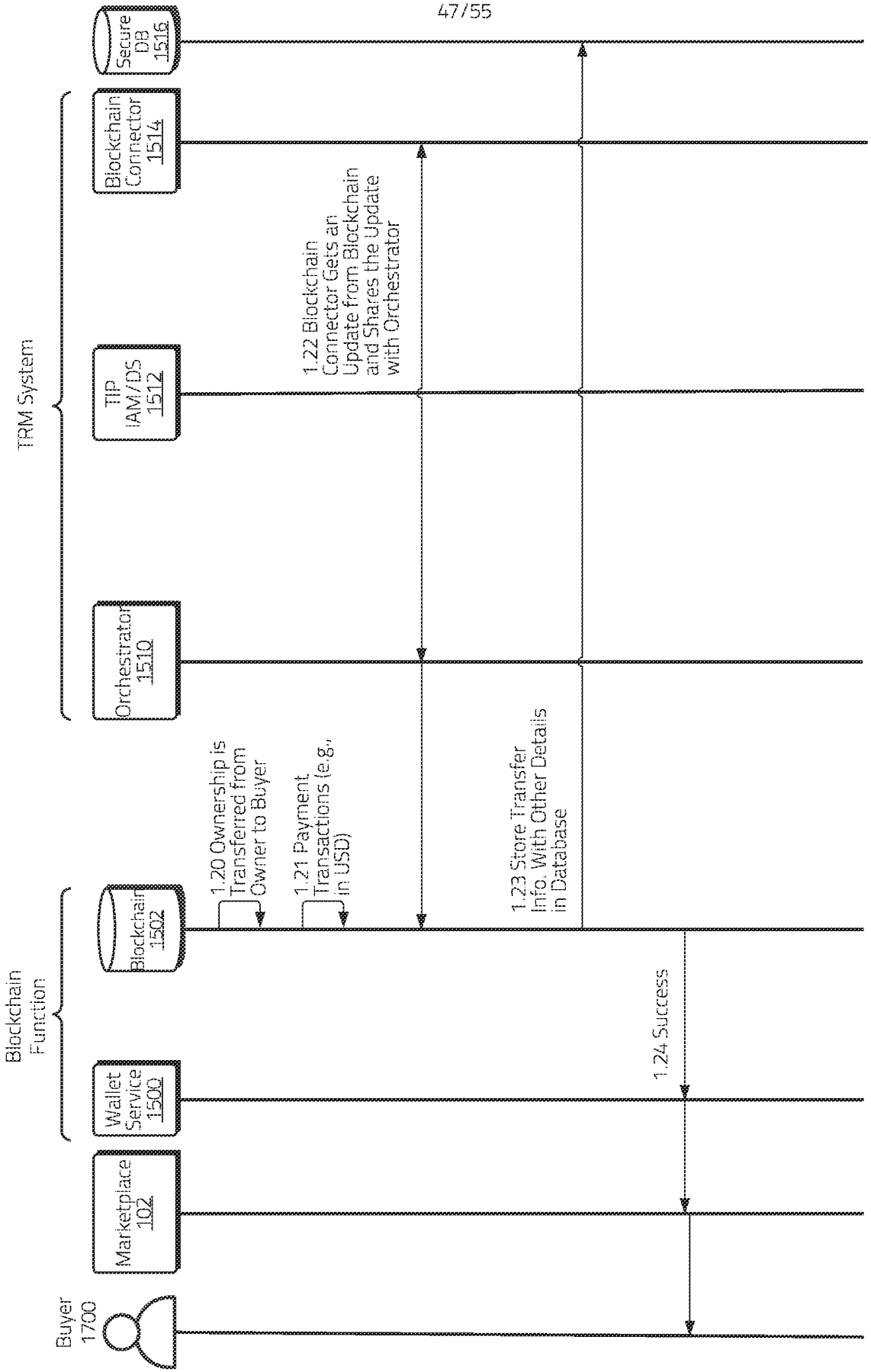

Buyer 1700
Marketplace 102
Wallet Service 1500
Blockchain Function
Blockchain 1502
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
Secure DB 1516
TRM System
1.20 Ownership is Transferred from Owner to Buyer
1.21 Payment Transactions (e.g., in USD)
1.22 Blockchain Connector Gets an Update from Blockchain and Shares the Update with Orchestrator
1.23 Store Transfer Info. With Other Details in Database
1.24 Success

Figure 17C

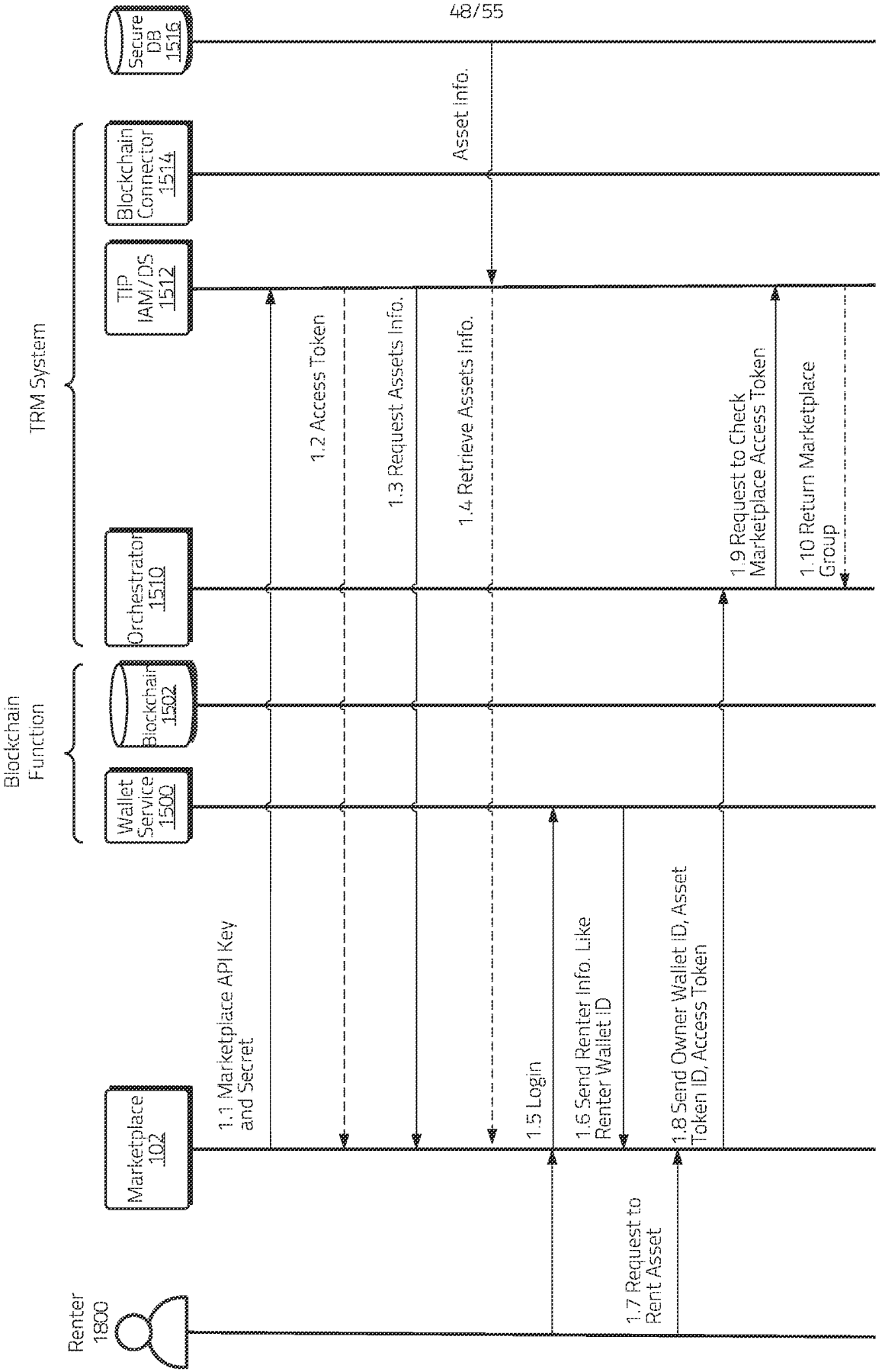

Renter
1800
Marketplace
102
Wallet
Service
1500
Blockchain
1502
Orchestrator
1510
TIP
IAM/DS
1512
Blockchain
Connector
1514
Secure
DB
1516
Blockchain
Function
TRM System
1.1 Marketplace API Key
and Secret
1.2 Access Token
1.3 Request Assets Info.
1.4 Retrieve Assets Info.
Asset Info.
1.5 Login
1.6 Send Renter Info. Like
Renter Wallet ID
1.7 Request to
Rent Asset
1.8 Send Owner Wallet ID, Asset
Token ID, Access Token
1.9 Request to Check
Marketplace Access Token
1.10 Return Marketplace
Group

Figure 18A

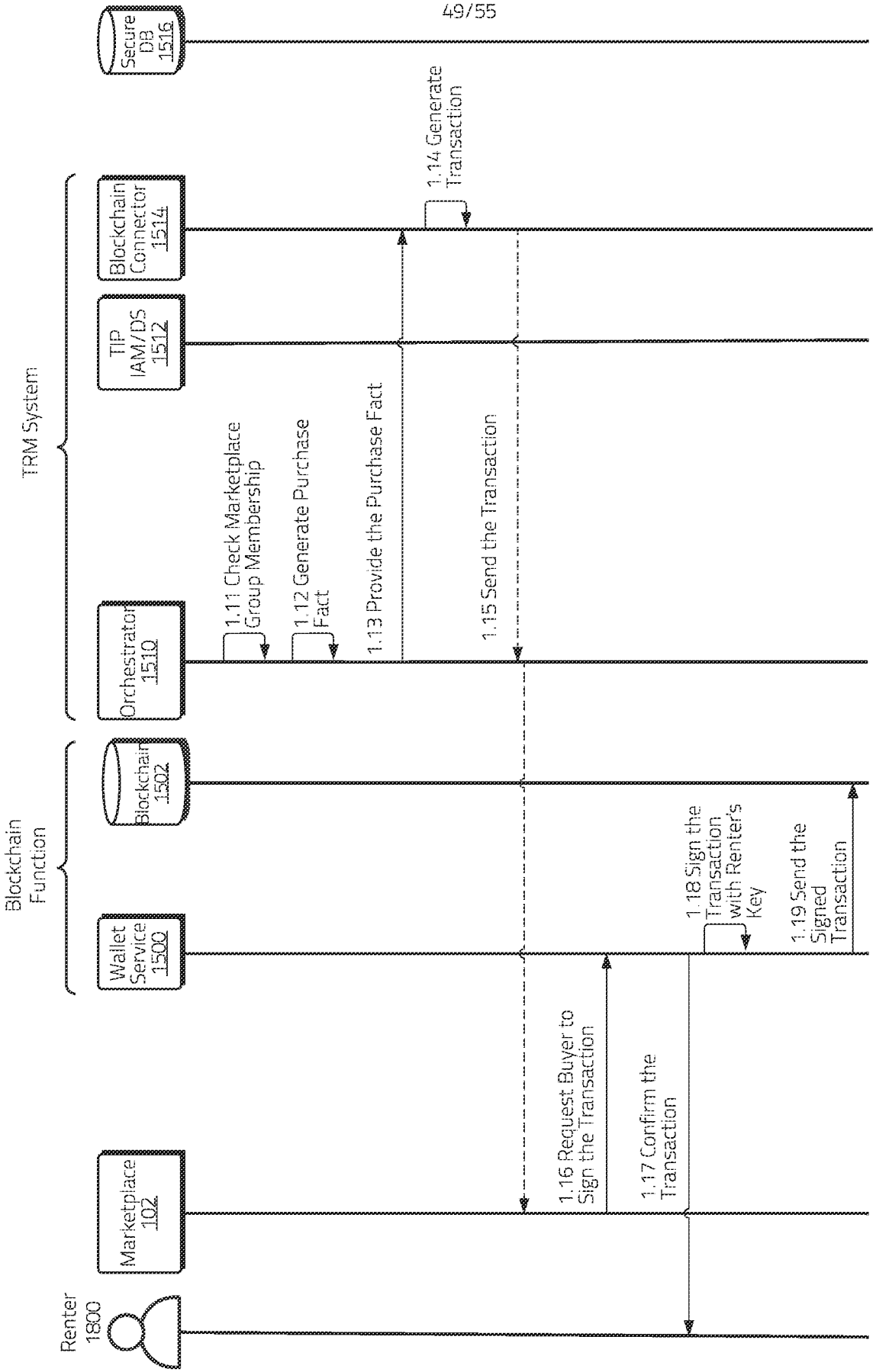

Renter
1800
Marketplace
102
Wallet
Service
1500
Blockchain
1502
Blockchain
Function
Orchestrator
1510
TIP
IAM/DS
1512
Blockchain
Connector
1514
Secure
DB
1516
TRM System
1.11 Check Marketplace
Group Membership
1.12 Generate Purchase
Fact
1.13 Provide the Purchase Fact
1.14 Generate
Transaction
1.15 Send the Transaction
1.16 Request Buyer to
Sign the Transaction
1.17 Confirm the
Transaction
1.18 Sign the
Transaction
with Renter's
Key
1.19 Send the
Signed
Transaction

Figure 18B

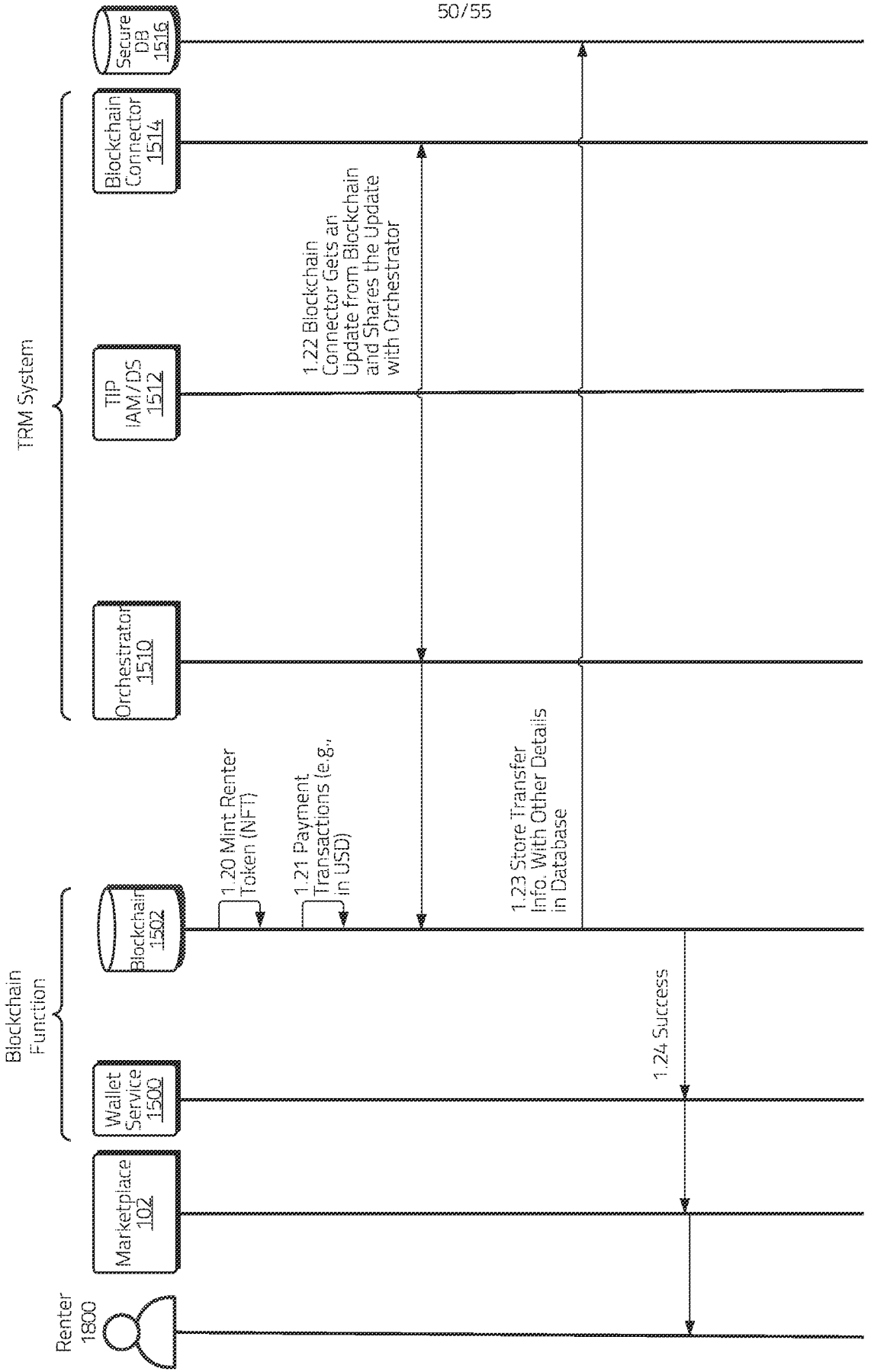

Renter
1800
Marketplace
102
Wallet
Service
1500
Blockchain
Function
Blockchain
1502
Orchestrator
1510
TIP
IAM/DS
1512
Blockchain
Connector
1514
Secure
DB
1516
TRM System
1.20 Mint Renter
Token (NFT)
1.21 Payment
Transactions (e.g.,
in USD)
1.22 Blockchain
Connector Gets an
Update from Blockchain
and Shares the Update
with Orchestrator
1.23 Store Transfer
Info. With Other Details
in Database
1.24 Success

Figure 18C

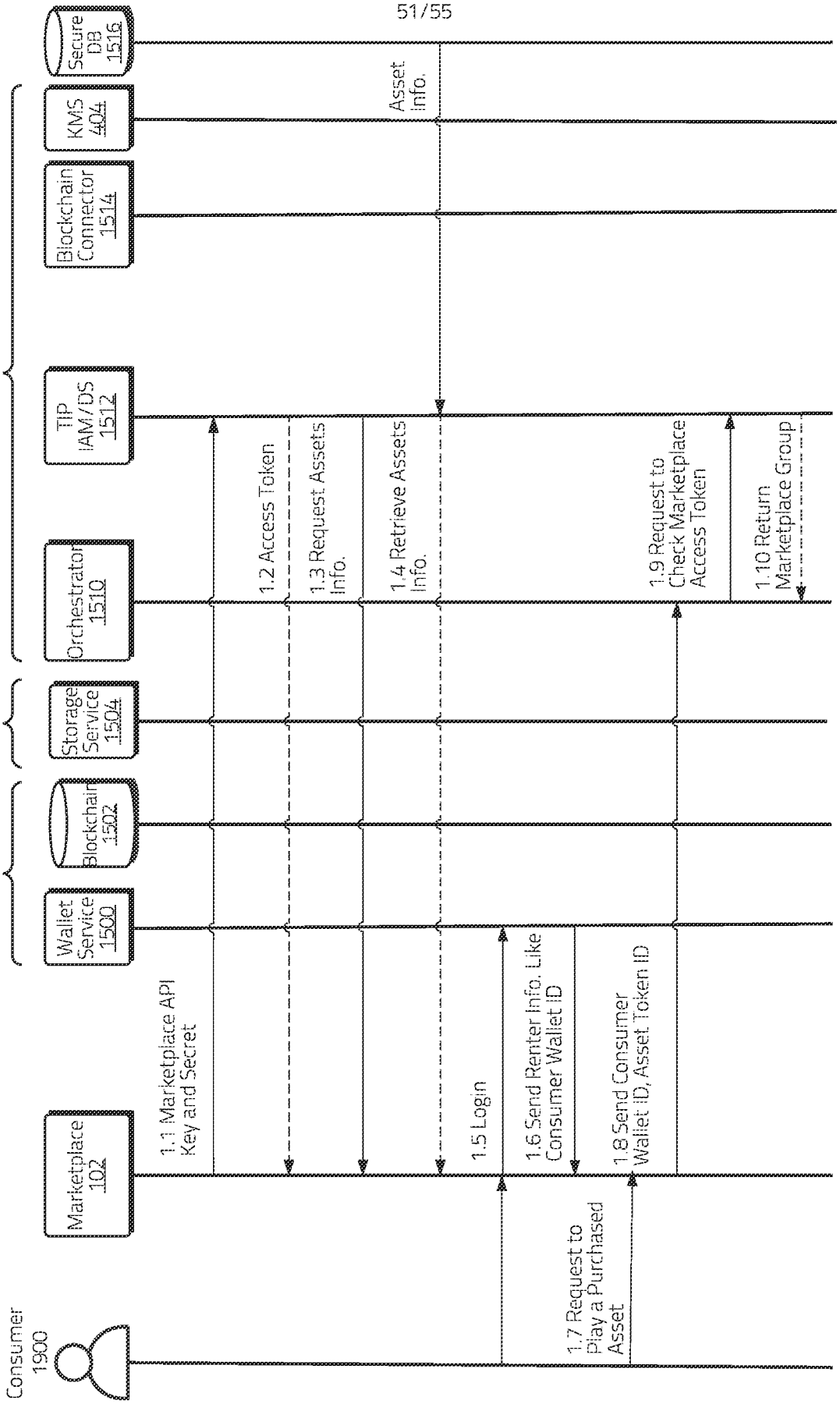

Consumer 1900
Marketplace 102
Wallet Service 1500
Blockchain 1502
Storage Service 1504
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
KMS 404
Secure DB 1516
Blockchain Function
Package Function
TRM System
1.1 Marketplace API Key and Secret
1.2 Access Token
1.3 Request Assets Info.
1.4 Retrieve Assets Info.
1.5 Login
1.6 Send Renter Info. Like Consumer Wallet ID
1.7 Request to Play a Purchased Asset
1.8 Send Consumer Wallet ID, Asset Token ID
1.9 Request to Check Marketplace Access Token
1.10 Return Marketplace Group
Asset Info.

Figure 19A

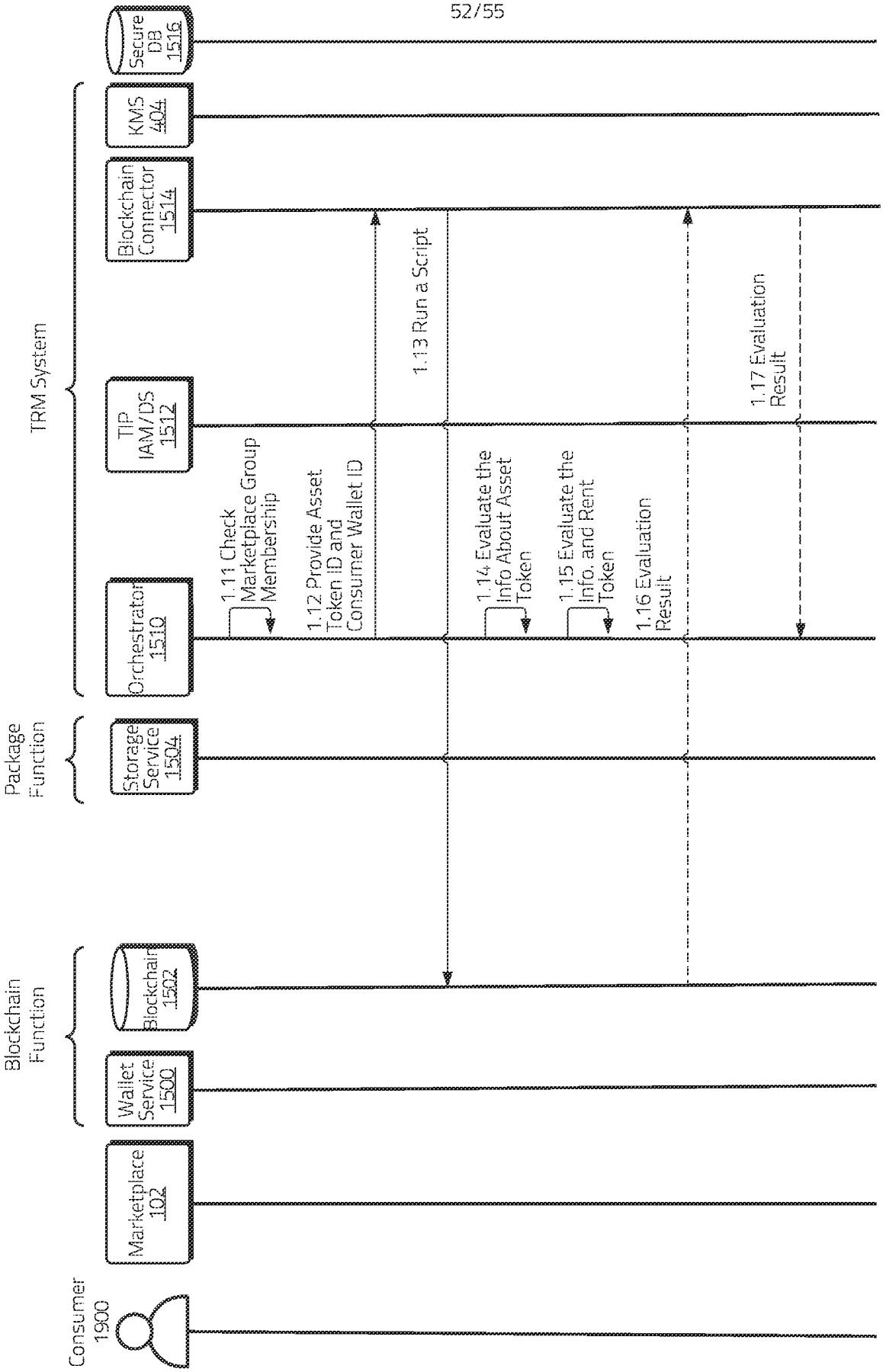

Consumer 1900
Marketplace 102
Wallet Service 1500
Blockchain 1502
Blockchain Function
Storage Service 1504
Package Function
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
KMS 404
Secure DB 1516
TRM System
1.11 Check Marketplace Group Membership
1.12 Provide Asset Token ID and Consumer Wallet ID
1.13 Run a Script
1.14 Evaluate the Info About Asset Token
1.15 Evaluate the Info. and Rent Token
1.16 Evaluation Result
1.17 Evaluation Result

Figure 19B

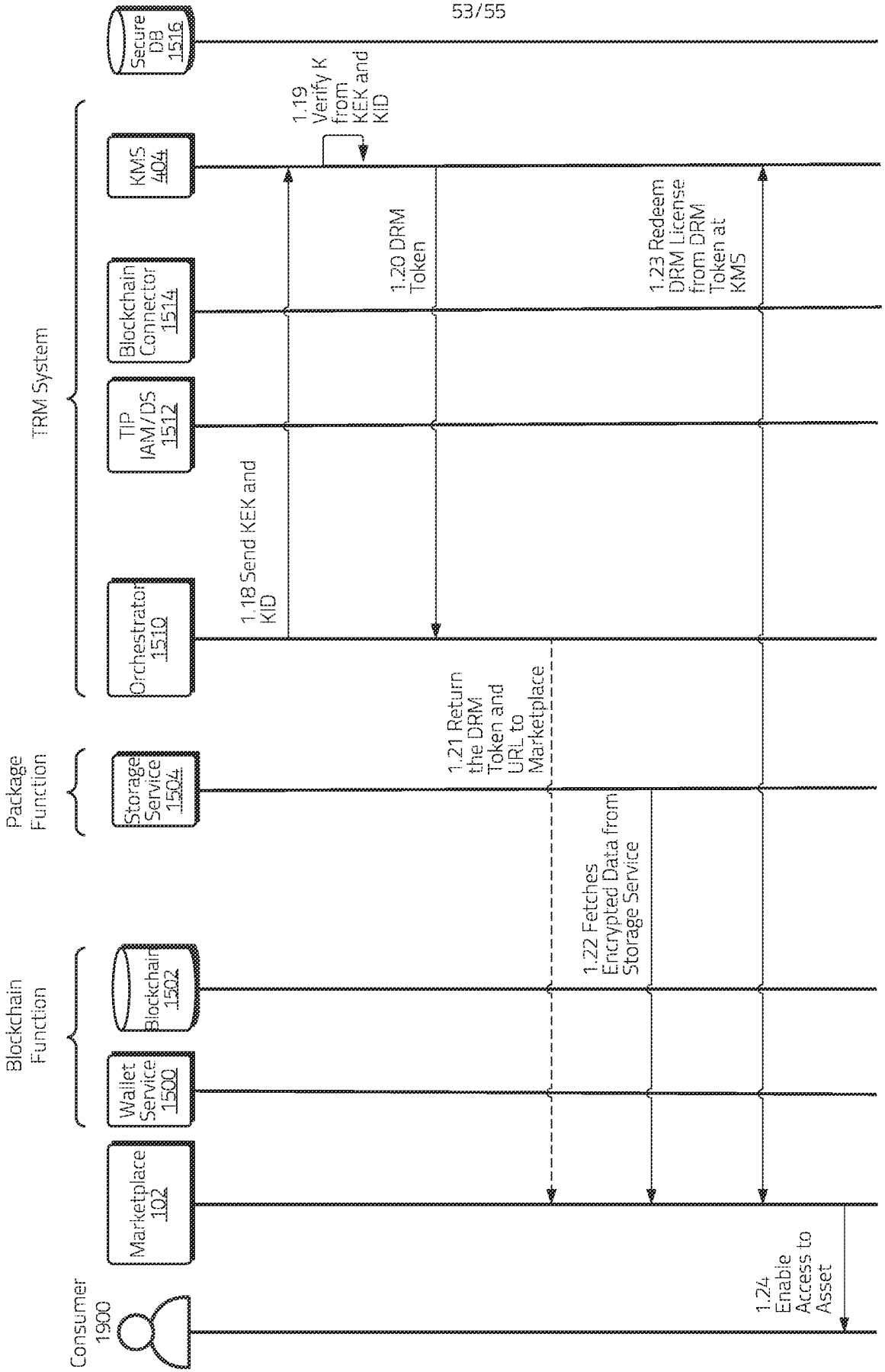

Consumer 1900
Blockchain Function
Package Function
TRM System
Marketplace 102
Wallet Service 1500
Blockchain 1502
Storage Service 1504
Orchestrator 1510
TIP IAM/DS 1512
Blockchain Connector 1514
KMS 404
Secure DB 1516
1.18 Send KEK and KID
1.19 Verify K from KEK and KID
1.20 DRM Token
1.21 Return the DRM Token and URL to Marketplace
1.22 Fetches Encrypted Data from Storage Service
1.23 Redeem DRM License from DRM Token at KMS
1.24 Enable Access to Asset

Figure 19C

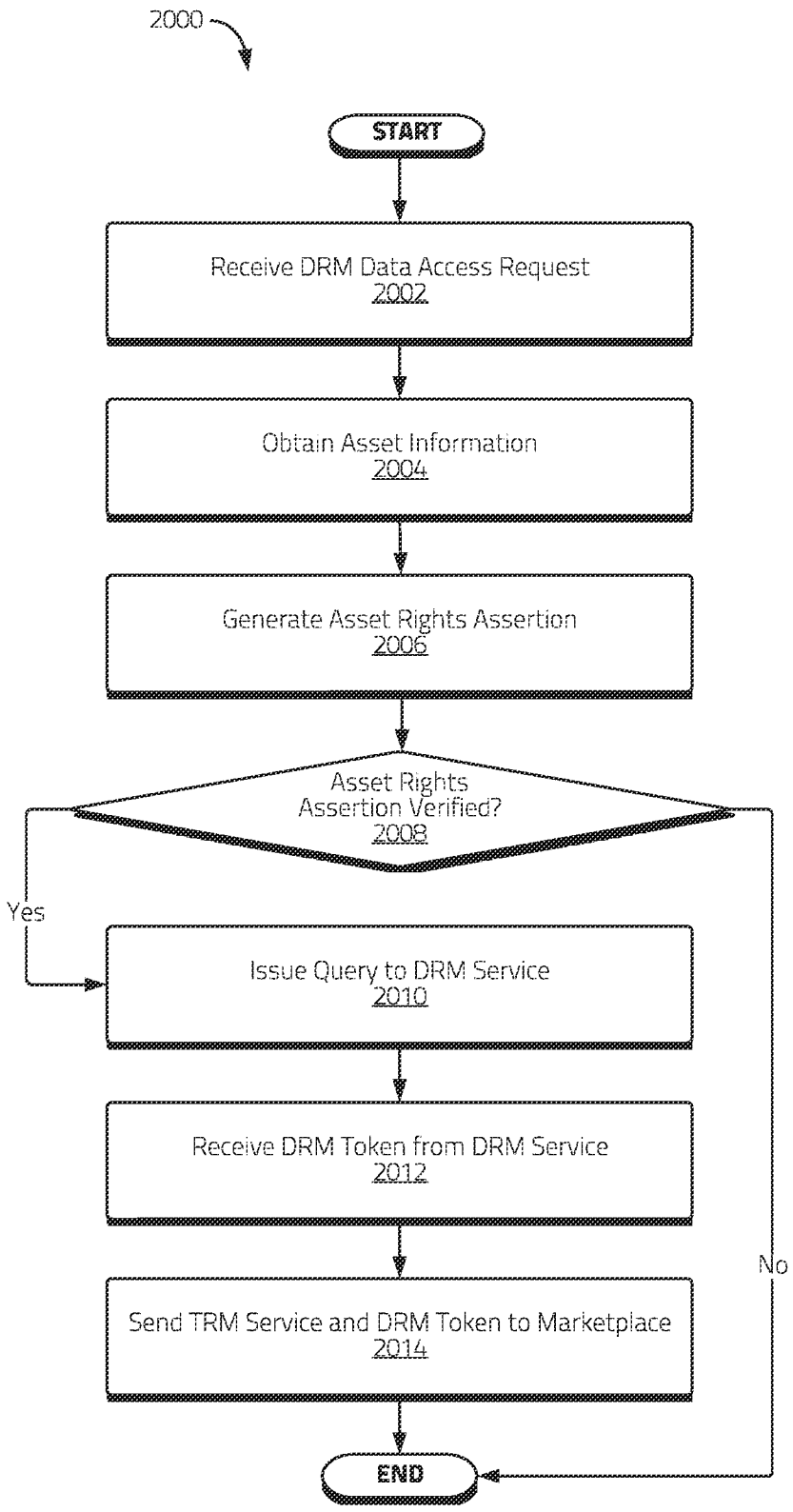

2000
START
Receive DRM Data Access Request
2002
Obtain Asset Information
2004
Generate Asset Rights Assertion
2006
Asset Rights Assertion Verified?
2008
Yes
No
Issue Query to DRM Service
2010
Receive DRM Token from DRM Service
2012
Send TRM Service and DRM Token to Marketplace
2014
END

Figure 20

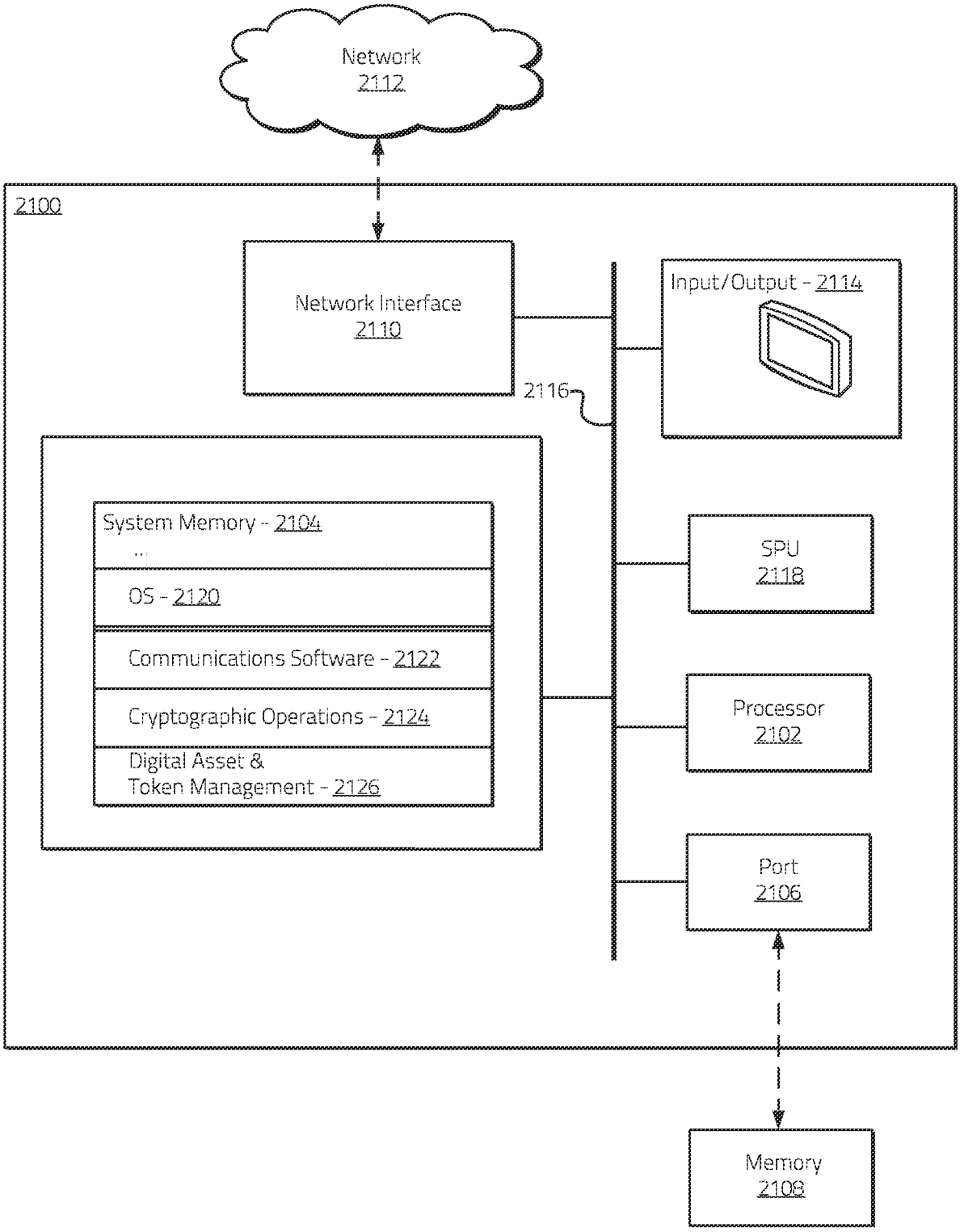

Network
2112
2100
Network Interface
2110
Input/Output - 2114
2116
System Memory - 2104
...
OS - 2120
Communications Software - 2122
Cryptographic Operations - 2124
Digital Asset &
Token Management - 2126
SPU
2118
Processor
2102
Port
2106
Memory
2108

Figure 21

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2022/045253

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F21/10 H04L9/40 H04W12/08 H04L9/32
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F H04L H04W H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2020/123464 A1 (SHELTERZOOM CORP [US]) 18 June 2020 (2020-06-18) figures 1-12 paragraphs [0002] - [0019] paragraphs [0038] - [0137] paragraphs [0160] - [0218] claims 1-61	1-16
A	US 2020/273048 A1 (ANDON CHRISTOPHER [US] ET AL) 27 August 2020 (2020-08-27) figures 1-13 paragraphs [0002] - [0025] paragraphs [0043] - [0145] claims 1-17	1-16

-/--

[x] Further documents are listed in the continuation of Box C.

[x] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance;; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance;; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 January 2023	20/01/2023

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Erdene-Ochir, O

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2022/045253

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2021/279305 A1 (GOLDSTON MARK [US] ET AL) 9 September 2021 (2021-09-09) figures 1-30 paragraphs [0004] - [0013] paragraphs [0048] - [0110] paragraphs [0112] - [0216] claims 1, 12, 17, 21-42	1-16
A	US 2021/035246 A1 (SCHOUPPE JIMMY [BE] ET AL) 4 February 2021 (2021-02-04) figures 1-9 paragraphs [0002] - [0035] paragraphs [0051] - [0091] claims 1-9, 12-20, 26-27	1-16

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2022/045253

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2020123464 A1	18-06-2020	CN 111902814 A	06-11-2020
		JP 7064651 B2	10-05-2022
		JP 2021518028 A	29-07-2021
		JP 2022009897 A	14-01-2022
		WO 2020123464 A1	18-06-2020
US 2020273048 A1	27-08-2020	US 2020273048 A1	27-08-2020
		US 2022188839 A1	16-06-2022
US 2021279305 A1	09-09-2021	US 2021279305 A1	09-09-2021
		US 2022215075 A1	07-07-2022
US 2021035246 A1	04-02-2021	US 2021035246 A1	04-02-2021
		WO 2021022000 A1	04-02-2021