(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년03월05일
(11) 등록번호 10-1370020
(24) 등록일자 2014년02월26일

(51) 국제특허분류(Int. Cl.)
G06F 21/53 (2013.01) *G06F 21/57* (2013.01)
G06F 15/16 (2006.01)
(21) 출원번호 10-2012-7033385
(22) 출원일자(국제) 2011년05월26일
심사청구일자 2013년02월15일
(85) 번역문제출일자 2012년12월21일
(65) 공개번호 10-2013-0033385
(43) 공개일자 2013년04월03일
(86) 국제출원번호 PCT/US2011/038133
(87) 국제공개번호 WO 2011/150204
국제공개일자 2011년12월01일
(30) 우선권주장
12/788,173 2010년05월26일 미국(US)
(56) 선행기술조사문헌
US20100023757 A1
US20090070582 A1
전체 청구항 수 : 총 53 항

(73) 특허권자
구글 인코포레이티드
미국 캘리포니아 마운틴 뷰 엠피시어터 파크웨이 1600 (우:94043)
(72) 발명자
브하누 헤먼트 매드오
미국 캘리포니아 94105 샌프란시스코 써드 프로워 폴섬 501
베이즈 루크
미국 캘리포니아 94103 샌프란시스코 나토마 스트리트 #4 555
밀스 앨런 스테판
미국 캘리포니아 94121 샌프란시스코 아파트먼트 3 풀톤 스트리트 6350
(74) 대리인
박장원

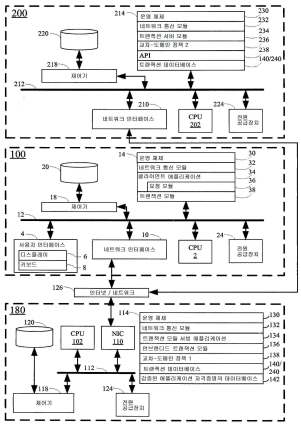
심사관 : 문남두

(54) 발명의 명칭 **보안 트랜잭션을 촉진하기 위하여 도메인-특정 보안 샌드박스를 사용하기 위한 시스템 및 방법**

(57) 요 약

보안 트랜잭션을 촉진하기 위한 컴퓨터 시스템, 방법, 및 컴퓨터 판독가능한 매체가 제공되며 여기서 클라이언트 애플리케이션은 클라이언트 컴퓨터 상에서 실행된다. 클라이언트 애플리케이션은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함하는 요청을 제1 도메인에 대하여 개시한다. 이러한 요청에 응답하여, 클라이언트는 검증된 트랜잭션 모듈을 제1 도메인으로부터 수신한다. 클라이언트 애플리케이션은 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리된 별도의 도메인 보안 샌드박스 내로 검증된 트랜잭션 모듈을 로딩한다. 검증된 트랜잭션 모듈은 제2 도메인과 검증된 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행한다. 별도로, 클라이언트 애플리케이션을 통하여, 제1 도메인을 질의함으로써 트랜잭션이 완료되었는지 여부에 대한 결정이 이루어진다.

대 표 도 - 도1



특허청구의 범위

청구항 1

보안 트랜잭션 촉진을 위한 컴퓨터 시스템에 있어서,

하나 이상의 처리 장치;

상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리

를 포함하며, 상기 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장하며, 상기 명령은

(A) 클라이언트 애플리케이션을 실행하는 명령, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청을 발생시키는 명령, 여기서 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(C) 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 명령;

(D) 상기 제출하는 명령(C)에 응답하여, 검증된 트랜잭션 모듈을 상기 제1 도메인으로부터 수신하는 명령, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 클라이언트 애플리케이션이 상기 검증된 트랜잭션 모듈을 실행하여, 상기 검증된 트랜잭션 모듈이 상기 메모리 내 별도의 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 별도의 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 별도의 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 명령(E)에 의해 실행되며, 그리고

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 검증된 트랜잭션 모듈이 별도의 도메인 보안 샌드박스에서 실행되는 동안 상기 검증된 트랜잭션 모듈로부터 제2 도메인으로 트랜잭션 호출을 전송하는 명령, 여기서 상기 제2 도메인은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 가짐;

(G) 상기 제2 도메인과 상기 검증된 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 명령; 및

(H) 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령

를 포함하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 2

제 1 항에 있어서, 상기 컴퓨터 시스템은 스크린 실제 영역을 갖는 디스플레이를 더욱 포함하며, 여기서 상기 실행하는 명령(A) 즉시, 상기 클라이언트 애플리케이션은 상기 스크린 실제 영역의 일부분 상에서 명시되고 상기 검증된 트랜잭션 모듈은 상기 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 3

제 1 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 4

제 3 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 5

제 1 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 6

제 1 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 컴퓨터 시스템에 접근할 수 있는 동일한 서버에 의해 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 7

제 1 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 컴퓨터 시스템에 각각 접근할 수 있는 별도의 서버에 의해 각각 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 8

제 1 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 검증된 트랜잭션 모듈은 상기 야기하는 명령(E) 동안 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 9

보안 트랜잭션 촉진 방법에 있어서,

(A) 프로그램된 컴퓨터 상에서 클라이언트 애플리케이션을 실행하는 단계, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 프로그램된 컴퓨터 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청을 발생시키는 단계, 여기서 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(C) 상기 프로그램된 컴퓨터로부터, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 단계;

(D) 상기 프로그램된 컴퓨터에서, 상기 제출하는 단계(C)에 응답하여, 검증된 트랜잭션 모듈을 상기 제1 도메인으로부터 수신하는 단계, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 검증된 트랜잭션 모듈을 실행하여, 상기 검증된 트랜잭션 모듈이 별도의 도메인 보안 샌드박스로 로딩되도록 야기하는 단계, 여기서

상기 별도의 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 별도의 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 단계(E)에 의해 실행되며, 그리고

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 검증된 트랜잭션 모듈이 상기 프로그램된 컴퓨터의 별도의 도메인 보안 샌드박스에서 실행되는 동안 상기 검증된 트랜잭션 모듈로부터 제2 도메인으로 트랜잭션 호출을 전송하는 단계, 여기서 상기 제2 도메인은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 가짐;

(G) 상기 프로그램된 컴퓨터를 사용하여, 상기 제2 도메인과 상기 검증된 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 단계; 및

(H) 상기 프로그램된 컴퓨터 상에서 실행중인 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 단계

를 포함하는, 보안 트랜잭션 촉진 방법.

청구항 10

제 9 항에 있어서, 상기 수행단계(A) 즉시, 상기 클라이언트 애플리케이션은 스크린 실제 영역의 일부분 상에서 명시되고 상기 검증된 트랜잭션 모듈은 상기 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시되는, 보안 트랜잭션 촉진 방법.

청구항 11

제 9 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진 방법.

청구항 12

제 11 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진 방법.

청구항 13

제 9 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진 방법.

청구항 14

제 9 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 동일한 서버에 의해 호스팅 되는, 보안 트랜잭션 촉진 방법.

청구항 15

제 9 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 별도의 서버에 의해 각각 호스팅 되는, 보안 트랜잭션 촉진 방법.

청구항 16

제 9 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 검증된 트랜잭션 모듈은 상기 야기 단계(E) 동안 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진 방법.

청구항 17

컴퓨터 시스템과 함께 사용되며, 컴퓨터 실행가능 명령이 수록된 비-일시적인(non-transitory) 컴퓨터 판독가능 매체로서,

상기 컴퓨터 실행 가능 명령은:

(A) 상기 컴퓨터 시스템 상에서 클라이언트 애플리케이션을 실행하는 명령, 여기서 상기 클라이언트 애플리케이

션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 컴퓨터 시스템 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청을 발생시키는 명령, 여기서 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(C) 상기 컴퓨터 시스템으로부터, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 명령;

(D) 상기 컴퓨터 시스템에서, 상기 제출하는 명령(C)에 응답하여, 검증된 트랜잭션 모듈을 상기 제1 도메인으로부터 수신하는 명령, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 컴퓨터 시스템을 사용해, 상기 클라이언트 애플리케이션이 상기 검증된 트랜잭션 모듈을 실행하여 상기 검증된 트랜잭션 모듈이 별도의 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 별도의 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 별도의 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 명령(E)에 의해 실행되며, 그리고

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 검증된 트랜잭션 모듈이 상기 컴퓨터 시스템의 별도의 도메인 보안 샌드박스에서 실행되는 동안 상기 검증된 트랜잭션 모듈로부터 제2 도메인으로 트랜잭션 호출을 전송하는 명령, 여기서 상기 제2 도메인은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 가짐;

(G) 상기 컴퓨터 시스템을 사용하여, 상기 제2 도메인과 상기 검증된 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 명령; 및

(H) 상기 컴퓨터 시스템 상에서 실행중인 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령

을 포함하는, 비-일시적인(non-transitory) 컴퓨터 판독가능 매체.

청구항 18

보안 트랜잭션 촉진을 위한 시스템에 있어서,

(A) 프로그램된 컴퓨터 상에서 클라이언트 애플리케이션을 실행하기 위한 수단, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 프로그램된 컴퓨터 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청을 발생시키기 위한 수단, 여기서 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(C) 상기 프로그램된 컴퓨터로부터, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하기 위한 수단;

(D) 상기 프로그램된 컴퓨터에서, 상기 제출(C)에 응답하여, 검증된 트랜잭션 모듈을 상기 제1 도메인으로부터 수신하는 수단, 여기서 상기 트랜잭션 모듈의 소스 URL이 상기 제1 도메인으로서 식별됨;

(E) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 검증된 트랜잭션 모듈을 실행하여 상기 검증된 트랜잭션 모듈이 별도의 도메인 보안 샌드박스로 로딩되도록 하는 수단, 여기서

상기 별도의 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 별도의 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 것(E)에 의해 실행되며, 그리고

상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 검증된 트랜잭션 모듈이 상기 프로그램된 컴퓨터의 별도의 도메인 보안 샌드박스에서 실행되는 동안 상기 검증된 트랜잭션 모듈로부터 제2 도메인으로 트랜잭션 호출을 전송하는 수단, 여기서 상기 제2 도메인은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 가짐;

(G) 상기 프로그램된 컴퓨터를 사용하여, 상기 제2 도메인과 상기 검증된 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 수단; 및

(H) 상기 프로그램된 컴퓨터 상에서 실행중인 상기 클라이언트 애플리케이션을 통하여 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하는 수단을

포함하며, 이에 의해 보안 트랜잭션을 촉진하도록 된 보안 트랜잭션 촉진을 위한 시스템.

청구항 19

보안 트랜잭션 촉진을 위한 컴퓨터 시스템에 있어서,

하나 이상의 처리 장치;

상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리

를 포함하며, 상기 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장하며,

여기서 상기 메모리는

무제한적인 제1 교차-도메인 정책에 의해 특징되는 제1 도메인;

제2 교차-도메인 정책에 의해 특징되는 제2 도메인, 여기서 상기 제2 교차-도메인 정책은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정함;

검증된 애플리케이션 자격증명의 데이터베이스;

상기 제1 도메인 및 상기 제2 도메인으로부터 판독 가능한 트랜잭션 데이터베이스; 및

(A) 상기 제1 도메인에서, 인터넷 또는 컴퓨터 네트워크를 통하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터 요청을 수신하는 명령, 여기서 상기 요청은 보안 인-애플리케이션 트랜잭션과 관련되며 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(B) 상기 클라이언트 애플리케이션에 대한 자격증명을 검증된 애플리케이션 자격증명의 데이터베이스에 대하여 확인하는 명령;

(C) 상기 요청을 상기 트랜잭션 데이터베이스 내로 키잉(keying)하는 명령;

(D) 상기 제1 도메인으로부터, 트랜잭션 모듈을 클라이언트 컴퓨터에 제공하는 명령;

(E) 상기 제2 도메인에서, 상기 클라이언트 컴퓨터 상에서 실행되는 상기 트랜잭션 모듈로부터 유래하는 트랜잭션 호출을 수신하는 명령, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제2 교차-도메인 정책을 따름;

(F) 상기 제2 도메인과 상기 클라이언트 컴퓨터 상에서 실행중인 상기 트랜잭션 모듈 사이의

검증된 트랜잭션을 수행하는 명령;

(G) 상기 제2 도메인에서, 완료된 트랜잭션의 기록을 트랜잭션 데이터베이스에 저장하는 명령;

(H) 상기 제1 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터, 상기 트랜잭션이 완료되었는지에 대한 질의를 수신하는 명령, 여기서 상기 질의는 요청을 고유하게 식별하는 트랜잭션 식별자를 포함함;

(I) 상기 제1 도메인에서, 상기 트랜잭션 데이터베이스 내에서 상기 트랜잭션 식별자를 검색함으로써, 상기 트랜잭션이 완료되었는지 여부를 결정하는 명령; 및

(J) 상기 수신하는 명령(H) 및 결정하는 명령(I)에 응답하여, 상기 클라이언트 컴퓨터 상에서 실행중인 상기 클라이언트 애플리케이션에게 상기 트랜잭션의 상태를 통보하는 명령

을 포함하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 20

제 19 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 21

제 20 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 22

제 19 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 23

제 19 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 클라이언트 컴퓨터에 접근할 수 있는 동일한 서버에 의해 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 24

제 19 항에 있어서,

상기 제1 도메인은 제1 서버에 의해 호스팅 되고,

상기 제2 도메인은 제2 서버에 의해 호스팅 되고,

상기 제1 서버 및 상기 제2 서버는 각각 인터넷 또는 컴퓨터 네트워크를 통하여 상기 클라이언트 컴퓨터에 접근 가능하며,

상기 메모리는 상기 제1 서버 내에 주재하는 제1 메모리 및 상기 제2 서버 내에 주재하는 제2 메모리를 포함하며,

상기 제1 교차-도메인 정책, 상기 검증된 애플리케이션 자격증명의 데이터베이스, 및 언브랜디드 트랜잭션 모듈은 상기 제1 서버의 제1 메모리에 주재하며,

상기 제2 교차-도메인 정책은 상기 제2 서버의 메모리에 주재하며, 그리고

상기 제1 서버 및 상기 제2 서버는 각각 트랜잭션 데이터베이스에 접근하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 25

제 19 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 트랜잭션 모듈은 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 26

제 19 항에 있어서, 상기 수신하는 명령(H), 결정하는 명령(I), 및 통보하는 명령(J)은 상기 보안 트랜잭션이 완료된 것으로 간주될 때까지 반복되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 27

보안 트랜잭션 촉진을 위한 컴퓨터 시스템에 있어서,

하나 이상의 처리 장치;

상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리

를 포함하며, 상기 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장하며, 상기 명령은

(A) 클라이언트 애플리케이션을 실행하는 명령, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제1 요청을 발생시키는 명령;

(C) 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제1 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 명령;

(D) 상기 제출명령(C)에 응답하여, 요청 모듈을 수신하는 명령, 여기서 상기 요청 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 클라이언트 애플리케이션이 상기 요청 모듈을 실행하여 상기 요청 모듈이 상기 메모리 내 제1 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 제1 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 제1 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 요청 모듈은 상기 요청 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 명령(E)에 의해 실행되며, 그리고

상기 요청 모듈은 상기 요청 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 요청 모듈을 통하여, 요청 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제2 요청을 발생시키는 명령, 여기서 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 제2 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(G) 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제2 요청을 인터넷 또는 컴퓨터 네트워크를 통하여, 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 갖는 제2 도메인에게 제출하는 명령;

(H) 상기 제출하는 명령(G)에 응답하여, 트랜잭션 모듈을 상기 제2 도메인으로부터 수신하는 명령, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제2 도메인으로서 식별됨;

(I) 상기 클라이언트 애플리케이션이 상기 트랜잭션 모듈을 실행하고 이에 따라 상기 트랜잭션 모듈이 상기 메모리 내 제2 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 제2 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 제2 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제2 도메인이라고 식별하는 프로

그램과 관련되고 여기에 한정되며,

상기 트랜잭션 모듈은 상기 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기 명령(I)에 의해 실행되며, 그리고

상기 트랜잭션 모듈은 상기 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(J) 상기 트랜잭션 모듈이 상기 제2 도메인 보안 샌드박스에서 실행되는 동안 상기 트랜잭션 모듈로부터 제3 도메인으로 트랜잭션 호출을 전송하는 명령, 여기서 상기 제3 도메인은 상기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정하는 교차-도메인 정책을 가짐;

(K) 상기 제3 도메인과 상기 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 명령; 및

(L) 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령

을 포함하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 28

제 27 항에 있어서, 상기 컴퓨터 시스템은 스크린 실제 영역을 갖는 디스플레이를 더욱 포함하며, 여기서 상기 실행하는 명령(A) 즉시, 상기 클라이언트 애플리케이션은 상기 스크린 실제 영역의 일부분 상에서 명시되고 상기 트랜잭션 모듈은 상기 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 29

제 27 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 30

제 29 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 31

제 27 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 32

제 27 항에 있어서, 상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 컴퓨터 시스템에 접근할 수 있는 동일한 서버에 의해 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 33

제 27 항에 있어서, 상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 컴퓨터 시스템에 각각 접근할 수 있는 별도의 서버에 의해 각각 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 34

제 27 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 트랜잭션 모듈은 상기 야기하는 명령(I) 동안 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 35

보안 트랜잭션 촉진 방법에 있어서,

(A) 프로그램된 컴퓨터 상에서, 클라이언트 애플리케이션을 실행하는 단계, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 프로그램된 컴퓨터 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제1 요청을 발생시키는 단계;

(C) 상기 프로그램된 컴퓨터를 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제1 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 단계;

(D) 상기 프로그램된 컴퓨터에서, 상기 제출하는 단계(C)에 응답하여, 요청 모듈을 수신하는 단계, 여기서 상기 요청 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 요청 모듈을 실행하여 상기 요청 모듈이 제1 도메인 보안 샌드박스로 로딩되도록 야기하는 단계, 여기서

상기 제1 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 제1 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 요청 모듈은 상기 요청 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기 단계(E)에 의해 실행되며, 그리고

상기 요청 모듈은 상기 요청 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 요청 모듈을 통하여, 상기 프로그램된 컴퓨터를 사용하여, 요청 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제2 요청을 발생시키는 단계, 여기서 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 제2 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(G) 상기 프로그램된 컴퓨터를 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제2 요청을 인터넷 또는 컴퓨터 네트워크를 통하여, 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 갖는 제2 도메인에게 제출하는 단계;

(H) 상기 제출하는 단계(G)에 응답하여, 상기 프로그램된 컴퓨터에서, 트랜잭션 모듈을 상기 제2 도메인으로부터 수신하는 단계, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제2 도메인으로서 식별됨;

(I) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 트랜잭션 모듈을 실행하여 상기 트랜잭션 모듈이 상기 메모리 내 제2 도메인 보안 샌드박스로 로딩되도록 야기하는 단계, 여기서

상기 제2 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 제2 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제2 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 트랜잭션 모듈은 상기 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 단계(I)에 의해 실행되며, 그리고

상기 트랜잭션 모듈은 상기 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(J) 상기 프로그램된 컴퓨터를 사용하여, 상기 트랜잭션 모듈이 상기 제2 도메인 보안 샌드박스에서 실행되는 동안 상기 트랜잭션 모듈로부터 제3 도메인으로 트랜잭션 호출을 전송하는 단계, 여기서 상기 제3 도메인은 상

기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정하는 교차-도메인 정책을 가짐;

(K) 상기 프로그램된 컴퓨터를 사용하여, 상기 제3 도메인과 상기 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 단계; 및

(L) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 단계

를 포함하는, 보안 트랜잭션 촉진 방법.

청구항 36

제 35 항에 있어서, 상기 실행하는 단계(A) 즉시, 상기 클라이언트 애플리케이션은 스크린 실제 영역의 일부분 상에서 명시되고 상기 트랜잭션 모듈은 상기 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시되는, 보안 트랜잭션 촉진 방법.

청구항 37

제 35 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진 방법.

청구항 38

제 37 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진 방법.

청구항 39

제 35 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진 방법.

청구항 40

제 35 항에 있어서, 상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인은 동일한 서버에 의해 호스팅되는, 보안 트랜잭션 촉진 방법.

청구항 41

제 37 항에 있어서, 상기 제1 도메인 및 상기 제2 도메인은 별도의 서버에 의해 각각 호스팅 되는, 보안 트랜잭션 촉진 방법.

청구항 42

제 37 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 트랜잭션 모듈은 상기 야기하는 단계(I) 동안 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진 방법.

청구항 43

컴퓨터 시스템과 함께 사용되며, 컴퓨터 실행가능 명령이 수록된 비-일시적인(non-transitory) 컴퓨터 판독가능 매체로서,

상기 컴퓨터 실행가능 명령은:

(A) 상기 컴퓨터 시스템 상에서, 클라이언트 애플리케이션을 실행하는 명령, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 컴퓨터 시스템 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제1 요청을 발생시키는 명령;

(C) 상기 컴퓨터 시스템을 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제1 요청을 인터넷 또는

컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하는 명령;

(D) 상기 컴퓨터 시스템에서, 상기 제출단계(C)에 응답하여, 요청 모듈을 수신하는 명령, 여기서 상기 요청 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 컴퓨터 시스템을 사용하여, 상기 클라이언트 애플리케이션이 상기 요청 모듈을 실행하여 상기 요청 모듈이 제1 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 제1 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 제1 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 요청 모듈은 상기 요청 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 명령(E)에 의해 실행되며, 그리고

상기 요청 모듈은 상기 요청 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 요청 모듈을 통하여, 상기 컴퓨터 시스템을 사용하여, 요청 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제2 요청을 발생시키는 명령, 여기서 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 제2 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(G) 상기 컴퓨터 시스템을 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제2 요청을 인터넷 또는 컴퓨터 네트워크를 통하여, 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 갖는 제2 도메인에게 제출하는 명령;

(H) 상기 제출하는 명령(G)에 응답하여, 상기 컴퓨터 시스템에서, 트랜잭션 모듈을 상기 제2 도메인으로부터 수신하는 명령, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제2 도메인으로서 식별됨;

(I) 상기 컴퓨터 시스템을 사용하여, 상기 클라이언트 애플리케이션이 상기 트랜잭션 모듈을 실행하여 상기 트랜잭션 모듈이 상기 메모리 내 제2 도메인 보안 샌드박스로 로딩되도록 야기하는 명령, 여기서

상기 제2 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 제2 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제2 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 트랜잭션 모듈은 상기 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 명령(I)에 의해 실행되며, 그리고

상기 트랜잭션 모듈은 상기 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(J) 상기 컴퓨터 시스템을 사용하여, 상기 트랜잭션 모듈이 상기 제2 도메인 보안 샌드박스에서 실행되는 동안 상기 트랜잭션 모듈로부터 제3 도메인으로 트랜잭션 호출을 전송하는 명령, 여기서 상기 제3 도메인은 상기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정하는 교차-도메인 정책을 가짐;

(K) 상기 컴퓨터 시스템을 사용하여, 상기 제3 도메인과 상기 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하는 명령; 및

(L) 상기 컴퓨터 시스템을 사용하여, 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령

을 포함하는, 비-일시적인(non-transitory) 컴퓨터 판독가능 매체.

청구항 44

보안 트랜잭션 촉진을 위한 시스템에 있어서,

(A) 프로그램된 컴퓨터 상에서, 클라이언트 애플리케이션을 실행하기 위한 수단, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행됨;

(B) 상기 프로그램된 컴퓨터 상에서, 상기 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제1 요청을 발생시키기 위한 수단;

(C) 상기 프로그램된 컴퓨터를 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제1 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하기 위한 수단;

(D) 상기 프로그램된 컴퓨터에서, 제출(C)에 응답하여, 요청 모듈을 수신하기 위한 수단, 여기서 상기 요청 모듈의 소스 URL은 상기 제1 도메인으로서 식별됨;

(E) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 요청 모듈을 실행하여 상기 요청 모듈이 제1 도메인 보안 샌드박스로 로딩되도록 야기하기 위한 수단, 여기서

상기 제1 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 메모리 스페이스로부터 격리되며,

상기 제1 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 요청 모듈은 상기 요청 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하기 위한 수단(E)에 의해 실행되며, 그리고

상기 요청 모듈은 상기 요청 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(F) 상기 요청 모듈을 통하여, 상기 프로그램된 컴퓨터를 사용하여, 요청 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제2 요청을 발생시키기 위한 수단; 여기서 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 제2 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함함;

(G) 상기 프로그램된 컴퓨터를 사용하여, 상기 보안 인-애플리케이션 트랜잭션에 대한 상기 제2 요청을 인터넷 또는 컴퓨터 네트워크를 통하여, 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 갖는 제2 도메인에게 제출하기 위한 수단;

(H) 상기 제출하기 위한 수단(G)에 응답하여, 상기 프로그램된 컴퓨터에서, 트랜잭션 모듈을 상기 제2 도메인으로부터 수신하기 위한 수단, 여기서 상기 트랜잭션 모듈의 소스 URL은 상기 제2 도메인으로서 식별됨;

(I) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션이 상기 트랜잭션 모듈을 실행하여 상기 트랜잭션 모듈이 상기 메모리 내 제2 도메인 보안 샌드박스로 로딩되도록 야기하기 위한 수단, 여기서

상기 제2 도메인 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며,

상기 제2 도메인 보안 샌드박스는 자신들의 소스 URL을 상기 제2 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정되며,

상기 트랜잭션 모듈은 상기 트랜잭션 모듈의 소스 URL의 신원이 변하거나 파괴되지 않도록 상기 야기하는 것(I)에 의해 실행되며, 그리고

상기 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 상기 클라이언트 애플리케이션에게 허용하지 않음;

(J) 상기 프로그램된 컴퓨터를 사용하여, 상기 트랜잭션 모듈이 상기 제2 도메인 보안 샌드박스에서 실행되는 동안 상기 트랜잭션 모듈로부터 제3 도메인으로 트랜잭션 호출을 전송하기 위한 수단, 여기서 상기 제3 도메인

은 상기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정하는 교차-도메인 정책을 가짐;

(K) 상기 프로그램된 컴퓨터를 사용하여, 상기 제3 도메인과 상기 트랜잭션 모듈 사이의 검증된 트랜잭션을 수행하기 위한 수단; 및

(L) 상기 프로그램된 컴퓨터를 사용하여, 상기 클라이언트 애플리케이션을 통하여, 상기 제1 도메인을 질의함으로써, 상기 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하기 위한 수단

을 포함하는, 보안 트랜잭션 촉진을 위한 시스템.

청구항 45

보안 트랜잭션 촉진을 위한 컴퓨터 시스템에 있어서,

하나 이상의 처리 장치;

상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리

를 포함하며, 상기 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장하며,

여기서 상기 메모리는

무제한적인 제1 교차-도메인 정책에 의해 특징되는 제1 도메인;

제2 교차-도메인 정책에 의해 특징되는 제2 도메인, 여기서 상기 제2 교차-도메인 정책은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정함;

제3 교차-도메인 정책에 의해 특징되는 제3 도메인, 여기서 상기 제3 교차-도메인 정책은 상기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정함;

검증된 애플리케이션 자격증명의 데이터베이스;

상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인으로부터 판독 가능한 트랜잭션 데이터베이스;

요청 모듈;

트랜잭션 모듈; 및

(A) 상기 제1 도메인에서, 인터넷 또는 컴퓨터 네트워크을 통하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터 제1 요청을 수신하는 명령, 여기서 상기 제1 요청은 보안 인-애플리케이션 트랜잭션과 관련됨;

(B) 상기 제1 도메인으로부터, 상기 요청 모듈을 상기 클라이언트 컴퓨터에 제공하는 명령;

(C) 상기 제2 도메인에서, 인터넷 또는 컴퓨터 네트워크을 통하여, 클라이언트 컴퓨터 상에서 실행중인 요청 모듈로부터 제2 요청을 수신하는 명령, 여기서 상기 제2 요청은 보안 인-애플리케이션 트랜잭션과 관련되며, 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택적으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함하며, 여기서 상기 요청 모듈의 소스 URL은 상기 제2 교차-도메인 정책에 따름;

(D) 상기 클라이언트 애플리케이션에 대한 자격증명을 검증된 애플리케이션 자격증명의 데이터베이스에 대하여 확인하는 명령;

(E) 상기 제2 요청을 상기 트랜잭션 데이터베이스 내로 키잉(keying)하는 명령;

(F) 상기 제2 도메인으로부터, 트랜잭션 모듈을 클라이언트 컴퓨터에 제공하는 명령;

(G) 상기 제3 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 트랜잭션 모듈로부터 유래한 트랜잭션 호출을 수신하는 명령, 여기서 검증된 트랜잭션 모듈의 소스 URL은 상기 제3 교차-도메인 정책에 따름.

(H) 상기 제3 도메인과 상기 클라이언트 컴퓨터 상에서 실행중인 상기 트랜잭션 모듈 사이의

검증된 트랜잭션을 수행하는 명령;

(I) 상기 제3 도메인에서, 완료된 트랜잭션의 기록을 트랜잭션 데이터베이스에 저장하는 명령;

(J) 상기 제1 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터, 상기 트랜잭션이 완료되었는지에 대한 질의를 수신하는 명령, 여기서 상기 질의는 요청을 고유하게 식별하는 트랜잭션 식별자를 포함함;

(K) 상기 제1 도메인에서, 상기 트랜잭션 데이터베이스 내에서 상기 트랜잭션 식별자를 검색함으로써, 상기 트랜잭션이 완료되었는지 여부를 결정하는 명령; 및

(L) 상기 수신하는 명령(J) 및 결정하는 명령(K)에 응답하여, 상기 클라이언트 컴퓨터 상에서 실행중인 상기 클라이언트 애플리케이션에게 상기 트랜잭션의 상태를 통보하는 명령;

을 포함하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 46

제 45 항에 있어서, 상기 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 47

제 46 항에 있어서, 상기 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 48

제 45 항에 있어서, 상기 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 49

제 45 항에 있어서, 상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 클라이언트 컴퓨터에 접근할 수 있는 동일한 서버에 의해 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 50

제 45 항에 있어서, 상기 제1 도메인, 상기 제2 도메인, 및 상기 제3 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 상기 컴퓨터 시스템에 각각 접근할 수 있는 별도의 서버에 의해 각각 호스팅 되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 51

제 45 항에 있어서,

상기 제1 도메인은 제1 서버에 의해 호스팅 되고,

상기 제2 도메인은 제2 서버에 의해 호스팅 되고,

상기 제3 도메인은 제3 서버에 의해 호스팅 되고,

상기 제1 서버, 상기 제2 서버, 및 상기 제3 서버는 각각 인터넷 또는 컴퓨터 네트워크를 통하여 상기 클라이언트 컴퓨터에 접근 가능하며,

상기 메모리는 상기 제1 서버 내에 주재하는 제1 메모리, 상기 제2 서버 내에 주재하는 내 제2 메모리, 및 상기 제3 서버 내에 주재하는 제3 메모리를 포함하며,

상기 제1 교차-도메인 정책 및 상기 요청 모듈은 상기 제1 서버의 제1 메모리에 주재하며,

상기 제2 교차-도메인 정책, 상기 검증된 애플리케이션 자격증명의 데이터베이스, 및 언브랜디드 트랜잭션 모듈은 상기 제2 서버의 제2 메모리에 주재하며, 그리고

상기 제3 교차-도메인 정책은 제3 서버의 메모리에 주재하는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 52

제 45 항에 있어서, 상기 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 상기 검증된 트랜잭션 모듈은 플래쉬(FLASH) SWF 애플리케이션인, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

청구항 53

제 45 항에 있어서, 상기 수신하는 명령(J), 결정하는 명령(K), 및 통보하는 명령(L)은 상기 보안 트랜잭션이 완료된 것으로 간주될 때까지 반복되는, 보안 트랜잭션 촉진을 위한 컴퓨터 시스템.

명 세 서

기 술 분 야

[0001] 본 출원은 일반적으로 보안 트랜잭션 촉진을 위한 도메인-특정 보안 샌드박스를 사용하는 시스템 및 방법에 관한 것이다.

배 경 기 술

[0002] 보안 실시간 트랜잭션 서비스를 요구하는 클라이언트 컴퓨터 상에서 실행되는 잠재적인 비보안 애플리케이션의 수가 지속적으로 증가하고 있다. 이러한 애플리케이션의 한 비-제한적 예는 플래쉬-기반 게이밍 애플리케이션(FLASH-based gaming application)인데 이는 레벨 해제와 같은 게임-내 업그레이드, 가상 장비, 가상 특수 무기 및 게이머에 대한 직접적인 치트(cheats directly to gamer)를 구입하기 위하여 가상 화폐의 재충전을 요구한다. 사용자 및 애플리케이션 개발자를 계정 정보의 부당 획득, 신분 도용, 및 또 다른 형태의 부정으로부터 보호하기 위하여 해당 분야에서 이러한 실시간 트랜잭션의 보안유지(securing)가 요구된다.

[0003] 이러한 트랜잭션을 보안유지하기 위한 한 가지 공지된 방법은 공유 비밀을 사용하는 개념이다(비밀 키 암호화(secret key cryptography)). 비밀 키 암호화는 싱글 키(single key)의 사용을 포함한다. 메시지와 키를 고려하면, 암호화는 이해 불가능한 데이터를 생성하며 이는 해독을 위한 키를 요구한다. 예컨대, 문헌 Section 2.4 of Kaufman, Network Security, Prentice-Hall, Inc., Upper Saddle River, N.J.를 참고하며, 이는 참조로서 본 명세서에 수록된다. 그렇지만, 공유 비밀 방법은 애플리케이션들 중 어느 하나가 보안유지 되지 않은 경우에는 작동하지 않는다. 예를 들어, 많은 대중적인 프로그래밍 애플리케이션은 플래쉬(FLASH) 플레이어에 의해 실행되고 보안유지 되지 않는다. 전형적으로, 공유 비밀 알고리즘이 사용될 때, 로컬 웹 서버를 호출하는 원격 웹 서버가 존재한다. 비밀은 원격 웹 서버 및 로컬 웹 서버 상에서 안전하며 이들 두 서버 사이에 통신되지 않는다. 이는, 클라이언트 컴퓨터에 다운로드 되고 예컨대 클라이언트 브라우저에서 실행되는 플래쉬(FLASH) 또는 또 다른 프로그램에 애플리케이션이 기록될 때 실패한다. 플래쉬(FLASH)의 경우, 사용자가 플래쉬(FLASH) 애플리케이션을 요구할 때, 플래쉬(FLASH) 플레이어에 의해 해석되는 바이트코드(bytecode)를 포함하는 SWF 파일이 클라이언트 컴퓨터에 다운로드 되고 클라이언트 브라우저 내에서 플래쉬(FLASH) 플레이어에 의해 실행(해석)된다. SWF 파일 내 바이트코드는 비밀을 결정하기 위하여 클라이언트 컴퓨터에서 검색될 수 있다. 따라서, 비밀은 플래쉬(FLASH) SWF 파일 내에 포함될 수 없다.

[0004] 전술한 배경을 고려하면, 해당 분야에서 요구되는 것은 보안유지 되지 않을 수 있는 애플리케이션으로부터 유래한 전자 트랜잭션을 인증하기 위한 개선된 시스템 및 방법이다.

발명의 내용

[0005] 본 발명은 서버 교차-도메인 정책(cross-domain policy)뿐만 아니라 도메인-특정 보안 샌드박스의 신규한 사용에 의해 해당 분야에서의 요구를 해결한다. 개시된 두 가지 구체 예가 있다. 첫 번째 구체 예에서, 클라이언트 애플리케이션은 무제한 교차-도메인 정책을 갖는 제1 도메인으로부터 트랜잭션 모듈을 요청한다. 일단 클라이언트 애플리케이션이 트랜잭션 모듈을 수신하면, 이는 트랜잭션 모듈의 소스 URL, 즉 제1 도메인의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스 내에서 실행된다. 트랜잭션 모듈은 소스 URL이 제1 도메인인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책을 갖는 제2 도메인과 상호대화함으로써 트랜잭션을 완료한다.

[0006] 두 번째 구체 예는 프로세스가 추가 단계를 수행한다. 두 번째 구체 예에서, 인-애플리케이션(in-application)

보안 트랜잭션을 수행하기 위한 필요성에 대한 응답으로, 클라이언트 애플리케이션은 인-애플리케이션 보안 트랜잭션과 관련된 제1 요청을 생성한다. 제1 요청은 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 교차-도메인 정책을 갖는 제1 도메인으로 전송된다. 이러한 요청에 응답하여, 제1 도메인은 클라이언트 애플리케이션으로 요청 모듈을 전송한다. 일단 클라이언트 애플리케이션이 요청 모듈을 수신하면, 이는 요청 모듈의 소스 URL, 즉 제1 도메인의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스(제1 샌드박스) 내에서 실행된다. 제1 샌드박스에서 작동하는 요청 모듈은 소스 URL이 제1 도메인인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책을 갖는 제2 도메인으로부터 트랜잭션 모듈을 요청한다. 제2 도메인은 클라이언트 애플리케이션으로 트랜잭션 모듈을 전송한다. 일단 클라이언트 애플리케이션이 트랜잭션 모듈을 수신하면, 트랜잭션 모듈은 트랜잭션 모듈의 소스 URL, 즉 제2 도메인의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스(제2 샌드박스) 내에서 실행된다. 트랜잭션 모듈은 소스 URL이 제2 도메인인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책을 갖는 제3 도메인과 상호대화함으로써 트랜잭션을 완료한다.

[0007] 교차-도메인 정책 개발에 의해, 그리고 내성(introspect)하기 위한 애플리케이션을 호출하기 위한 파워 없이 자신의 도메인-특정 보안 샌드박스 내에서 프로그램을 실행시키는 본질적인 능력에 의해, 본 발명은 보안 인-애플리케이션 트랜잭션을 촉진하기 위한, 고도로 보안 유지된 시스템, 방법, 및 컴퓨터 판독가능한 매체를 제공한다.

[0008] *클라이언트 관점에서의 첫 번째 구체 예.*

[0009] 클라이언트 관점에서 본 발명의 첫 번째 구체 예의 한 가지 실시는 보안 트랜잭션을 촉진하기 위한 컴퓨터 시스템을 포함한다. 컴퓨터 시스템은 하나 이상의 처리 장치 및 상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리를 포함한다. 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장한다. 예시적인 목적을 위하여, 이러한 컴퓨터 시스템은 클라이언트 컴퓨터로서 간주되는데 여기서 클라이언트 애플리케이션은 컴퓨터 시스템과 관련된 로컬 데이터로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행된다. 본 구체 예에서, 보안 인-애플리케이션 트랜잭션과 관련된 요청은 클라이언트 애플리케이션이 실행되는 시간에, 클라이언트 애플리케이션을 통하여 발생된다. 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명(credential), (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자(identifier), 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디(identification)를 포함한다. 보안 인-애플리케이션 트랜잭션에 대한 요청은 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에 제출된다. 이러한 제출에 응답하여, 검증된(validated) 트랜잭션 모듈이 제1 도메인으로부터 수신된다. 이에 따라, 트랜잭션 모듈의 소스 URL가 제1 도메인으로서 식별된다.

[0010] 클라이언트 애플리케이션이 상기 검증된 트랜잭션 모듈을 실행하고 이에 따라 상기 검증된 트랜잭션 모듈이 컴퓨터 시스템의 메모리 내 별도의 도메인-특정 보안 샌드박스로 로딩된다. 상기 별도의 도메인-특정 보안 샌드박스는 상기 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리된다. 별도의 도메인-특정 보안 샌드박스는 자신들의 소스 URL을 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정된다. 상기 검증된 트랜잭션 모듈의 소스 URL의 신원(identity)이 변경되거나 파기되지 않도록 상기 검증된 트랜잭션 모듈이 실행된다. 더욱이, 상기 검증된 트랜잭션 모듈은 상기 검증된 트랜잭션 모듈을 내성하는 파워를 클라이언트 애플리케이션에게 허용하지 않는다.

[0011] 검증된 트랜잭션 모듈은, 별도의 도메인-특정 보안 샌드박스 내에서 실행되는 동안, 트랜잭션 호출(call)을 제2 도메인으로 보낸다. 제2 도메인은 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 가진다. 검증된 트랜잭션은 제2 도메인과 상기 검증된 트랜잭션 모듈 사이에서 수행된다.

[0012] 명령은 앞서-제시된 프로세스 중 임의 것 또는 모든 것과 함께 수행되거나 또는 앞서-제시된 프로세스 모두가 수행된 이후에 수행되는 명령을 더욱 포함한다. 이러한 명령은 클라이언트 애플리케이션을 통하여, 제1 도메인을 질의(query)함으로써, 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령을 포함한다.

[0013] 일부 경우에서, 컴퓨터 시스템은 스크린 실제 영역(screen real estate)을 갖는 디스플레이를 더욱 포함한다. 이러한 일부 경우에서, 클라이언트 애플리케이션은 스크린 실제 영역의 일부분 상에서 명시되고(manifested) 상기 검증된 트랜잭션 모듈은 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시된다. 이는 클라이언트 애플리케이션의 사용자에게 인-애플리케이션 트랜잭션이 클라이언트 애플리케이션 내에서부터 실행되는 심리스(seamless) 트랜잭션이라는 인상을 유리하게 제공한다.

[0014] 일부 경우에, 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임(in-game) 트랜잭션이다. 일부 경우에, 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매이다. 일부 경우에, 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션이다.

[0015] 일부 경우에, 제1 도메인 및 제2 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 컴퓨터 시스템에 접근할 수 있는 동일한 서버에 의해 호스팅 된다. 또 다른 경우, 제1 도메인 및 제2 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 컴퓨터 시스템에 각각 접근할 수 있는 별도의 서버에 의해 각각 호스팅 된다.

[0016] 일부 경우에, 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 검증된 트랜잭션 모듈은 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션이다.

[0017] *서버 관점에서의 첫 번째 구체 예.*

[0018] 본 발명은 또한 클라이언트를 서비스하는 하나 이상의 서버의 관점에서 앞서-제시된 첫 번째 구체 예를 고려한다. 예를 들어, 이러한 서버 관점의 한 가지 실시는 하나 이상의 처리 장치 및 상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리를 포함하는 컴퓨터 시스템을 제공한다. 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장한다. 메모리는 무제한적인 제1 교차-도메인 정책에 의해 특징되는 제1 도메인, 제2 교차-도메인 정책에 의해 특징되는 제2 도메인 여기서 상기 제2 교차-도메인 정책은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정함, 검증된 애플리케이션 자격증명의 데이터베이스, 제1 도메인 및 제2 도메인으로부터 판독 가능한 트랜잭션 데이터베이스, 및 언브랜디드(unbranded) 트랜잭션 모듈을 포함한다.

[0019] 일부 경우에, 컴퓨터 시스템은 제1 컴퓨터 및 제2 컴퓨터를 포함하고 앞서-제시된 메모리는 제1 컴퓨터 내에 주재하는 메모리 및 제2 컴퓨터 내에 주재하는 메모리를 포함한다. 이러한 경우, 제1 교차-도메인 정책, 검증된 애플리케이션 자격증명의 데이터베이스, 및 언브랜디드 트랜잭션 모듈은 제1 컴퓨터의 메모리에 주재할 수 있으며 한편 제2 교차-도메인 정책은 제2 컴퓨터의 메모리에 주재할 수 있다. 또한, 이러한 구체 예에서, 트랜잭션 데이터베이스에 대한 접근은 제1 컴퓨터 및 제2 컴퓨터로부터 가능하다. 일부 대안적인 경우, 컴퓨터 시스템은 단일 컴퓨터이다.

[0020] 메모리는 제1 도메인에서, 인터넷 또는 컴퓨터 네트워크를 통하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터 요청을 수신하기 위한 명령을 포함한다. 상기 요청은 보안 인-애플리케이션 트랜잭션과 관련된다. 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 클라이언트 애플리케이션의 사용자의 아이디, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함한다. 메모리는 검증된 애플리케이션 자격증명의 데이터베이스에 대하여 클라이언트 애플리케이션에 대한 자격증명을 확인하기 위한 명령을 포함한다. 메모리는 트랜잭션 데이터베이스 내로 요청을 키잉(keying)하기 위한 명령을 더욱 포함한다. 메모리는 검증된 트랜잭션 모듈을 다이내믹하게 발생시키기 위한 명령을 더욱 포함한다. 한 구체 예에서, 하나 이상의 자격증명이 언브랜디드 트랜잭션 모듈 내로 주입(inject)된다. 이러한 주입(보안) 방법의 예는 예를 들어, 2009.10.27. 출원된 미국 특허 출원 일련번호 12/607,005, 명칭 "Systems and Methods for Authenticating an Electronic Transaction"에서 발견되며, 이는 그 전체가 참조로서 본 명세서에 수록된다. 본 명세서에 기재된 또 다른 구체 예에서, 또 다른 방법이 트랜잭션 모듈을 제공하는 도메인으로부터 파라미터를 획득하기 위하여 사용되며 이러한 파라미터들은 트랜잭션 모듈을 검증하는 역할을 한다.

[0021] 메모리는 제1 도메인으로부터, 검증된 트랜잭션 모듈을 클라이언트 컴퓨터에 제공하기 위한 명령을 더욱 포함한다. 메모리는 제2 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 검증된 트랜잭션 모듈로부터 유래하는 트랜잭션 호출을 수신하기 위한 명령을 더욱 포함하며, 여기서 검증된 트랜잭션 모듈의 소스 URL은 제2 교차-도메인 정책에 따른다. 메모리는 제2 도메인과 클라이언트 컴퓨터 상에서 실행되는 검증된 트랜잭션 모듈 사이에서, 검증된 트랜잭션을 수행하기 위한 명령을 더욱 포함한다. 메모리는 제2 도메인에서, 트랜잭션 데이터베이스 내에 완료된 트랜잭션의 기록을 저장하기 위한 명령을 더욱 포함한다.

[0022] 전술한 프로세스의 일부 또는 모두와 동시에, 또는 전술한 프로세스 모두가 완료된 이후, 추가 프로세스가 수행된다. 이러한 프로세스는 제1 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터, 트랜잭션이 완료되었는지에 대한 질의(query)를 수신하는 것을 포함한다. 상기 질의는 요청을 고유하게 식

별하는 트랜잭션 식별자를 포함한다. 이러한 프로세스는 제1 도메인에서, 트랜잭션 데이터베이스 내에서 트랜잭션 식별자를 검색(looking up)함으로써, 트랜잭션이 완료되었는지 여부를 결정하는 것을 더욱 포함한다. 이러한 프로세스는 상기 수신 및 결정에 응답하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션에게 트랜잭션 상태를 통보하는 것을 더욱 포함한다.

[0023] 컴퓨터 시스템이 제1 서버 및 제2 서버로 분할되는 구체 예에서, 제1 도메인에서 또는 제1 도메인에 의해 수행되는 앞서-제시된 프로세스는 제1 서버에 의해 완료되며 한편 제2 도메인에서 또는 제2 도메인에 의해 수행되는 앞서-제시된 프로세스는 제2 서버에 의해 완료된다. 해당 분야의 통상의 기술자는 제1 도메인이 복수의 제1 컴퓨터를 포함할 수 있고 제2 도메인이 복수의 제2 컴퓨터를 포함할 수 있음을 이해할 것이다. 예컨대, 하나의 서버는 도메인 내에서 다른 서버에 대하여 거울상 위치일 수 있다. 또 다른 예에서, 하나의 서버는 도메인 내에서 다른 서버의 백업(backup) 서버일 수 있다. 또 다른 예에서, 하나의 도메인은 종래 부하균등기법(load balancing technique)을 통하여 공통 부하(load)를 조작하는 복수의 서버를 포함할 수 있다. 해당 분야의 통상의 기술자는 본 발명이 이러한 서로 다른 모든 양상을 완전하게 고려한다는 것을 이해할 것이다.

[0024] *클라이언트 관점의 두 번째 구체 예.*

[0025] 클라이언트 관점에서 본 발명의 두 번째 구체 예의 한 가지 실시는 하나 이상의 처리 장치 및 상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리를 포함하는 컴퓨터 시스템을 포함한다. 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장한다. 명령은 클라이언트 애플리케이션을 실행하기 위한 명령을 포함하며, 여기서 상기 클라이언트 애플리케이션은 로컬 데이터 스토어로부터 직접 실행되거나 또는 원격 클라이언트 애플리케이션 서버로부터 실행된다. 명령은 클라이언트 애플리케이션을 통하여, 상기 클라이언트 애플리케이션이 실행중인 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제1 요청을 발생시키기 위한 명령을 포함한다. 명령은 보안 인-애플리케이션 트랜잭션에 대한 제1 요청을 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책을 갖는 제1 도메인에게 제출하기 위한 명령을 포함한다. 명령은 상기 제출에 응답하여, 요청 모듈을 수신하기 위한 명령을 더욱 포함하며, 여기서 상기 요청 모듈의 소스 URL은 제1 도메인으로서 식별된다.

[0026] 명령은 클라이언트 애플리케이션이 요청 모듈을 실행하도록 하여 이에 따라 요청 모듈이 메모리 내의 제1 도메인-특정 보안 샌드박스로 로딩되도록 하기 위한 명령을 더욱 포함한다. 제1 도메인-특정 보안 샌드박스는 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리된다. 제1 도메인-특정 보안 샌드박스는 자신들의 소스 URL을 제1 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정된다. 요청 모듈의 소스 URL의 신원이 변경되거나 파기되지 않도록 요청 모듈이 실행된다. 요청 모듈은 상기 요청 모듈을 내성하는 파워를 클라이언트 애플리케이션에게 허용하지 않는다.

[0027] 명령은 요청 모듈을 통하여, 요청 애플리케이션이 실행되는 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 제2 요청을 발생시키기 위한 명령을 포함한다. 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 제2 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함한다. 명령은 보안 인-애플리케이션 트랜잭션에 대한 제2 요청을, 인터넷 또는 컴퓨터 네트워크를 통하여, 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 제1 도메인인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책을 갖는 제2 도메인에게 제출하기 위한 명령을 더욱 포함한다. 명령은 상기 제출에 응답하여, 제2 도메인으로부터 트랜잭션 모듈을 수신하기 위한 명령을 더욱 포함하며, 여기서 상기 트랜잭션 모듈의 소스 URL은 제2 도메인으로서 식별된다.

[0028] 명령은 클라이언트 애플리케이션이 트랜잭션 모듈을 실행하도록 하여 이에 따라 트랜잭션 모듈이 상기 메모리 내의 제2 도메인-특정 보안 샌드박스로 로딩되도록 하기 위한 명령을 더욱 포함한다. 제2 도메인-특정 보안 샌드박스는 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되며, 제2 도메인-특정 보안 샌드박스는 자신들의 소스 URL을 제2 도메인이라고 식별하는 프로그램과 관련되고 여기에 한정된다. 트랜잭션 모듈의 소스 URL의 신원이 변경되거나 파기되지 않도록 상기와 같이 트랜잭션 모듈이 실행된다. 트랜잭션 모듈은 검증된 트랜잭션 모듈을 내성하는 파워를 클라이언트 애플리케이션에게 허용하지 않는다.

[0029] 명령은 트랜잭션 모듈로부터, 상기 트랜잭션 모듈이 제2 도메인-특정 보안 샌드박스 내에서 실행되는 동안, 트랜잭션 호출을 제3 도메인으로 보내기 위한 명령을 더욱 포함한다. 제3 도메인은 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 제2 도메인인 이러한 외부 프로그램으로 한정하는 교차-도메인 정책을 가진다. 명령은 검증된 트랜잭션을 제3 도메인과 상기 트랜잭션 모듈 사이에서 수행시키기 위한 명령을 포함한다.

[0030] 명령은 앞서-제시된 프로세스 중 임의 것 또는 모든 것과 함께 수행되거나 또는 앞서-제시된 프로세스 모두가 수행된 이후에 수행되는 명령을 더욱 포함한다. 이러한 명령은 클라이언트 애플리케이션을 통하여, 제1 도메인을 질의(query)함으로써, 트랜잭션이 완료되었는지를 결정하고, 이에 따라 보안 트랜잭션을 촉진하는 명령을 포함한다.

[0031] 일부 경우에서, 컴퓨터 시스템은 스크린 실제 영역을 갖는 디스플레이를 더욱 포함하며, 여기서 클라이언트 애플리케이션의 실시 즉시, 클라이언트 애플리케이션은 스크린 실제 영역의 일부분 상에서 명시된다. 이러한 경우, 상기 검증된 트랜잭션 모듈은 스크린 실제 영역의 상기 일부분의 서브셋 상에서 명시된다.

[0032] 일부 경우에, 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션이다. 일부 경우에, 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매이다. 일부 경우에, 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션이다.

[0033] 일부 경우에, 제1, 제2, 및 제3 도메인은 각각 인터넷 또는 컴퓨터 네트워크를 통하여 컴퓨터 시스템에 접근할 수 있는 동일한 서버에 의해 호스팅 된다. 일부 경우에, 제1, 제2, 및 제3 도메인은 각각 인터넷 또는 컴퓨터 네트워크를 통하여 컴퓨터 시스템에 각각 접근할 수 있는 별도의 서버에 의해 호스팅 된다. 일부 경우에, 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 검증된 트랜잭션 모듈은 상기 클라이언트 애플리케이션에 의해 로딩되는 플래쉬(FLASH) SWF 애플리케이션이다.

[0034] *서버 관점에서의 두 번째 구체 예.*

[0035] 본 발명은 또한 클라이언트를 서비스하는 하나 이상의 서버의 관점에서 앞서-제시된 두 번째 구체 예를 고려한다. 예를 들어, 이러한 서버 관점의 한 가지 실시는 보안 트랜잭션을 촉진하기 위한 컴퓨터 시스템을 제공한다. 컴퓨터 시스템은 하나 이상의 처리 장치 및 상기 하나 이상의 처리 장치 중 적어도 하나에 연결된 메모리를 포함한다. 메모리는 상기 하나 이상의 처리 장치 중 적어도 하나에 의해 실행되는 명령을 저장한다.

[0036] 메모리는 무제한적인 제1 교차-도메인 정책에 의해 특징되는 제1 도메인, 제2 교차-도메인 정책에 의해 특징되는 제2 도메인 여기서 상기 제2 교차-도메인 정책은 상기 제2 도메인과 상기 제2 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제1 도메인인 이러한 외부 프로그램으로 한정함, 제3 교차-도메인 정책에 의해 특징되는 제3 도메인을 포함하며 여기서 상기 제3 교차-도메인 정책은 상기 제3 도메인과 상기 제3 도메인의 외부에 있는 프로그램 사이의 상호대화를 자신의 소스 URL이 상기 제2 도메인인 이러한 외부 프로그램으로 한정한다. 상기 메모리는 검증된 애플리케이션 자격증명의 데이터베이스, 제1 도메인, 제2 및 제3 도메인으로부터 판독 가능한 트랜잭션 데이터베이스, 언브랜디드 트랜잭션 모듈, 및 요청 모듈을 더욱 포함한다.

[0037] 일부 경우에, 컴퓨터 시스템은 제1, 제2, 및 제3 컴퓨터를 포함하고 앞서-제시된 메모리는 제1 컴퓨터 내에 주재하는 메모리, 제2 컴퓨터 내 주재하는 메모리, 및 제3 컴퓨터 내에 주재하는 메모리를 포함한다. 이러한 경우, 제1 교차-도메인 정책 및 요청 모듈은 제1 컴퓨터의 메모리에 주재할 수 있다. 제2 교차-도메인 정책, 검증된 애플리케이션 자격증명의 데이터베이스, 및 언브랜디드 트랜잭션 모듈은 제2 컴퓨터의 메모리에 주재할 수 있다. 제3 교차-도메인 정책은 제2 컴퓨터의 메모리에 주재할 수 있다. 또한, 이러한 구체 예에서, 트랜잭션 데이터베이스에 대한 접근은 제1 컴퓨터, 제2 컴퓨터, 및 제3 컴퓨터로부터 가능하다. 일부 대안적인 경우, 컴퓨터 시스템은 단일 컴퓨터이다.

[0038] 메모리는 제1 도메인에서, 인터넷 또는 컴퓨터 네트워크를 통하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터 제1 요청을 수신하기 위한 명령을 포함하며, 여기서 상기 제1 요청은 보안 인-애플리케이션 트랜잭션과 관련된다. 메모리는 제1 도메인으로부터, 요청 모듈을 클라이언트 컴퓨터에게 제공하기 위한 명령을 더욱 포함한다. 메모리는 제2 도메인에서, 인터넷 또는 컴퓨터 네트워크를 통하여, 클라이언트 컴퓨터 상에서 실행중인 요청 모듈로부터 제2 요청을 수신하기 위한 명령을 포함하며, 여기서 상기 제2 요청은 보안 인-애플리케이션 트랜잭션과 관련되며 상기 제2 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자, 및 (iii) 선택사항으로, 클라이언트 애플리케이션의 사용자의 아이디를 포함한다. 이러한 경우, 요청 모듈의 소스 URL은 제2 교차-도메인 정책에 따른다.

[0039] 메모리는 검증된 애플리케이션 자격증명의 데이터베이스에 대하여 클라이언트 애플리케이션에 대한 자격증명을 확인하기 위한 명령을 더욱 포함한다. 메모리는 트랜잭션 데이터베이스 내로 제2 요청을 키잉(keying)하기 위한 명령을 더욱 포함한다. 메모리는 검증된 트랜잭션 모듈을 다이내믹하게 발생시키기 위한 명령을 더욱 포함한다.

예컨대, 일부 구체 예에서, 검증된 트랜잭션은 하나 이상의 자격증명을 언브랜디드 트랜잭션 모듈에 주입시킴으로써 발생된다. 이러한 주입(보안) 방법의 예는 예를 들어, 2009.10.27. 출원된 미국 특허 출원 일련번호 12/607,005, 명칭 "Systems and Methods for Authenticating an Electronic Transaction"에서 발견되며, 이는 그 전체가 참조로서 본 명세서에 수록된다. 본 명세서에 기재된 또 다른 구체 예에서, 또 다른 방법이 트랜잭션 모듈을 제공하는 도메인으로부터 파라미터를 획득하기 위하여 사용되며 이러한 파라미터들은 트랜잭션 모듈을 검증하는 역할을 한다. 메모리는 제2 도메인으로부터, 검증된 트랜잭션 모듈을 클라이언트 컴퓨터에 제공하기 위한 명령을 더욱 포함한다. 메모리는 제3 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 검증된 트랜잭션 모듈로부터 유래하는 트랜잭션 호출을 수신하기 위한 명령을 더욱 포함하며, 여기서 검증된 트랜잭션 모듈의 소스 URL은 제3 교차-도메인 정책에 따른다. 메모리는 제3 도메인과 클라이언트 컴퓨터 상에서 실행되는 검증된 트랜잭션 모듈 사이에서, 검증된 트랜잭션을 수행하기 위한 명령을 더욱 포함한다. 메모리는 제3 도메인에서, 트랜잭션 데이터베이스 내에 완료된 트랜잭션의 기록을 저장하기 위한 명령을 더욱 포함한다.

[0040] 전술한 프로세스의 일부 또는 모두와 동시에, 또는 전술한 프로세스 모두가 완료된 이후, 추가 프로세스가 수행된다. 이러한 프로세스는 제1 도메인에서, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션으로부터, 트랜잭션이 완료되었는지에 대한 질의를 수신하는 것을 포함한다. 상기 질의는 요청을 고유하게 식별하는 트랜잭션 식별자를 포함한다. 이러한 프로세스는 제1 도메인에서, 트랜잭션 데이터베이스 내에서 트랜잭션 식별자를 검색함으로써, 트랜잭션이 완료되었는지 여부를 결정하는 것을 더욱 포함한다. 이러한 프로세스는 상기 수신 및 결정에 응답하여, 클라이언트 컴퓨터 상에서 실행중인 클라이언트 애플리케이션에게 트랜잭션의 상태를 통보하는 것을 더욱 포함한다.

[0041] 일부 경우에, 보안 인-애플리케이션 트랜잭션은 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션이다. 일부 경우에, 인-게임 업그레이드는 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매이다. 일부 경우에, 클라이언트 애플리케이션은 소셜 네트워킹 애플리케이션, 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션이다.

[0042] 일부 경우에, 제1, 제2, 및 제3 도메인은 인터넷 또는 컴퓨터 네트워크를 통하여 클라이언트 컴퓨터에 접근할 수 있는 동일한 서버에 의해 호스팅 된다.

[0043] 일부 경우에, 제1 도메인은 제1 서버에 의해 호스팅 되고, 제2 도메인은 제2 서버에 의해 호스팅 되고, 그리고 제3 도메인은 제3 서버에 의해 호스팅 된다. 제1, 제2 및 제3 서버는 각각 인터넷 또는 컴퓨터 네트워크를 통하여 클라이언트 컴퓨터에 접근 가능하다. 이러한 일부 경우, 앞서-제시된 메모리는 제1 서버 내에 주재하는 제1 메모리, 제2 서버 내에 주재하는 제2 메모리, 및 제3 서버 내에 주재하는 제3 메모리를 포함한다. 이러한 일부 경우, 제1 교차-도메인 정책 및 요청 모듈은 제1 서버의 제1 메모리에 주재한다. 이러한 일부 경우, 제2 교차-도메인 정책, 검증된 애플리케이션 자격증명의 데이터베이스, 및 언브랜디드 트랜잭션 모듈은 제2 서버의 제2 메모리에 주재한다. 이러한 일부 경우, 제3 교차-도메인 정책은 제3 서버의 메모리에 주재한다.

[0044] 일부 경우에, 클라이언트 애플리케이션은 플래쉬(FLASH) 애플리케이션이고 검증된 트랜잭션 모듈은 플래쉬(FLASH) SWF 애플리케이션이다.

[0045] 컴퓨터 시스템이 제1, 제2, 및 제3 서버로 분할되는 구체 예에서, 제1 도메인에서 또는 제1 도메인에 의해 수행되는 앞서-제시된 프로세스는 제1 서버에 의해 완료되며, 제2 도메인에서 또는 제2 도메인에 의해 수행되는 앞서-제시된 프로세스는 제2 서버에 의해 완료되며, 그리고 제3 도메인에서 또는 제3 도메인에 의해 수행되는 앞서-제시된 프로세스는 제3 서버에 의해 완료된다. 해당 분야의 통상의 기술자는 제1 도메인이 복수의 제1 컴퓨터를 포함할 수 있고, 제2 도메인이 복수의 제2 컴퓨터를 포함할 수 있고, 그리고 제3 도메인이 복수의 제3 컴퓨터를 포함할 수 있음을 이해할 것이다. 예컨대, 하나의 서버는 도메인 내에서 다른 서버에 대하여 거울상 위치일 수 있다. 또 다른 예에서, 하나의 서버는 도메인 내에서 다른 서버의 백업(backup) 서버일 수 있다. 또 다른 예에서, 하나의 도메인은 종래 부하균등기법(load balancing technique)을 통하여 공통 부하(load)를 조작하는 복수의 서버를 포함할 수 있다. 해당 분야의 통상의 기술자는 본 발명이 이러한 서로 다른 모든 양상을 완전하게 고려한다는 것을 이해할 것이다.

도면의 간단한 설명

[0046] 도 1은 본 발명의 첫 번째 구체 예에 따르는 시스템을 나타낸다.

도 2A 및 2B는 본 발명의 첫 번째 구체 예에 따르는 방법을 나타낸다.

도 3A 및 3B는 본 발명의 두 번째 구체 예에 따르는 시스템을 나타낸다.

도 4A 및 4B는 본 발명의 첫 번째 구체 예에 따르는 방법을 나타낸다.

유사한 도면 부호는 도면의 일부 관점 전반에서 대응하는 부품을 의미한다.

발명을 실시하기 위한 구체적인 내용

[0047] 본 발명은 잠재적으로 보안유지 되지 않는 애플리케이션에 의해 통신되는 전자 트랜잭션을 인증하기 위한 공지된 시스템 및 방법에 비하여 신규한 개선점을 상세하게 기술한다. 본 발명은 서버 교차-도메인 정책뿐만 아니라 도메인-특정 보안 샌드박스를 사용한다. 개시된 두 가지 구체 예가 있다. 도 1 및 2에 제시된 첫 번째 구체 예에서, 클라이언트 애플리케이션(34)은 무제한 교차-도메인 정책(138)을 갖는 제1 도메인(180)으로부터 트랜잭션 모듈(38)을 요청한다. 일단 클라이언트 애플리케이션(34)이 트랜잭션 모듈(38)을 수신하면, 이는 트랜잭션 모듈의 소스 URL, 즉 제1 도메인(180)의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스 내에서 실행된다. 트랜잭션 모듈(38)은 소스 URL이 제1 도메인(180)인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책(236)을 갖는 제2 도메인(200)과 상호대화함으로써 트랜잭션을 완료한다.

[0048] 도 3 및 4에 제시된 두 번째 구체 예는 프로세스가 추가 단계를 수행한다. 두 번째 구체 예에서, 인-애플리케이션 보안 트랜잭션을 수행하기 위한 필요성에 대한 응답으로, 클라이언트 애플리케이션(34B)은 인-애플리케이션 보안 트랜잭션과 관련된 제1 요청을 생성한다. 제1 요청은 인터넷 또는 컴퓨터 네트워크(302)를 통하여 무제한 교차-도메인 정책(336)을 갖는 제1 도메인(300)(도 3B)으로 전송된다. 이러한 요청에 응답하여, 제1 도메인(300)은 클라이언트 애플리케이션(34B)으로 요청 모듈(36)을 전송한다. 일단 클라이언트 애플리케이션(34B)이 요청 모듈(36)을 수신하면, 상기 요청 모듈(36)은 요청 모듈(36)의 소스 URL, 즉 제1 도메인(300)의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스(제1 샌드박스) 내에서 실행된다. 제1 샌드박스에서 작동하는 요청 모듈(36)은 소스 URL이 제1 도메인(300)인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책(138B)을 갖는 제2 도메인(180B)으로부터 트랜잭션 모듈(38)을 요청한다.

[0049] 일부 구체 예에서, 클라이언트 애플리케이션(34B)으로부터 트랜잭션 모듈에 대한 요청을 수신한 즉시, 제2 도메인(180B)은 보안 정보를 트랜잭션 모듈(136)의 언브랜디드 버전 내로 주입시키고, 이에 따라 보안 트랜잭션 모듈(38)을 형성하고, 클라이언트 애플리케이션(34B)으로 보안 트랜잭션 모듈(38)을 전송한다. 일단 클라이언트 애플리케이션(34B)이 보안 트랜잭션 모듈(38)을 수신하면, 트랜잭션 모듈(38)은 보안 트랜잭션 모듈(38)의 소스 URL, 즉 제2 도메인(180)의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스(제2 샌드박스) 내에서 실행된다. 트랜잭션 모듈(38)은 소스 URL이 제2 도메인(180)인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책(236)을 갖는 제3 도메인(200)과 상호대화함으로써 트랜잭션을 완료한다.

[0050] 일부 구체 예에서, 제2 도메인(180B)은 트랜잭션 모듈(136)의 내용의 변형 없이 트랜잭션 모듈(136)의 언브랜디드 버전을 제공한다. 환언하면, 일부 구체 예에서, 자격 증명이 트랜잭션 모듈(136)의 언브랜디드 버전으로 반드시 주입되어 이에 의해 보안 트랜잭션 모듈(38)을 형성하는 것은 아니다. 이러한 구체 예에서, 일단 요청 모듈(36)이 언브랜디드 트랜잭션 모듈(136)을 수신하면, 이는 제2 도메인(180B)에 의해 제공된 파라미터로 트랜잭션 모듈(136)을 검증할 수 있다. 이러한 방식으로, 요청 모듈(36)은 트랜잭션 모듈(136)을 검증할 수 있다(이에 의해 파라미터를 트랜잭션 모듈(136)로 주입하지 않으면서, 트랜잭션 모듈(136)이 트랜잭션 모듈(38)로 간주될 수 있다). 그 후, 트랜잭션 모듈(38)은 보안 트랜잭션 모듈(38)의 소스 URL, 즉 제2 도메인(180B)의 URL이 보존되도록 자신의 도메인-특정 보안 샌드박스(제2 샌드박스) 내에서 실행된다. 트랜잭션 모듈(38)은 소스 URL이 제2 도메인(180B)인 이러한 프로그램 및 프로세스와의 상호대화를 제한하는 교차-도메인 정책(236)을 갖는 제3 도메인(200)과 상호대화함으로써 트랜잭션을 완료한다.

[0051] 교차-도메인 정책 개발에 의해, 그리고 내성하기 위한 애플리케이션을 호출하기 위한 파워 없이 자신의 도메인-특정 보안 샌드박스 내에서 프로그램을 실행시키는 본질적인 능력에 의해, 본 발명은 보안 인-애플리케이션 트랜잭션을 촉진하기 위한, 고도로 보안 유지된 시스템, 방법, 및 컴퓨터 판독가능한 매체를 제공한다.

[0052] 보안 인-애플리케이션 트랜잭션을 수행하기 위한 신규한 시스템 및 방법의 개요가 개시된 지금, 본 발명의 첫 번째 구체 예에 따르는 시스템의 더욱 상세한 설명을 도 1과 결합하여 개시한다. 이에 따라, 도 1은 본 발명에 따른 환경의 토폴로지를 나타낸다.

[0053] 토폴로지에서, 보안 인터페이스 서버(180), 클라이언트 장치(100), 및 트랜잭션 서버(200)가 존재한다. 물론,

또 다른 토폴로지도 가능하다. 예컨대, 보안 인터페이스 서버(180)는 실제로 수 개의 서버를 포함할 수 있다. 더욱이, 전형적으로, 수백, 수천, 수십만 개 또는 그 이상의 클라이언트 장치(100)가 존재한다. 도 1에 제시된 예시적인 토폴로지는 단지 해당 분야의 통상의 기술자에게 용이하게 이해되고자 하는 방식으로 본 발명의 첫 번째 구체 예의 특징을 설명하는 역할을 한다.

[0054] 보안 인터페이스 서버(180)는 전형적으로 하나 이상의 처리 장치(CPU)(102), 네트워크 또는 또 다른 통신 인터페이스(110), 메모리(114), 선택적으로 하나 이상의 제어기(118)에 의해 접속되는 하나 이상의 자기 디스크 저장 및/또는 영구 장치(120), 전술한 부품들을 상호연결시키기 위한 하나 이상의 통신 버스(112), 및 전술한 부품들에게 전원 공급하기 위한 전원 공급장치(124)를 포함할 것이다. 메모리(114) 내 데이터는 캐싱(caching)과 같은 공지된 연산 기법을 사용하여 비-휘발성 메모리(120)와 심리스하게(seamlessly) 공유될 수 있다. 메모리(114) 및/또는 메모리(120)는 중앙 처리 장치(102)와 관련하여 원격으로 위치한 대용량 저장장치를 포함할 수 있다. 환언하면, 메모리(114) 및/또는 메모리(120)에 저장된 일부 데이터는 실제로 보안 인터페이스 서버(180) 외부에 있지만 네트워크 인터페이스(110)를 사용하는 인터넷, 인트라넷, 또는 또 다른 형태의 네트워크 또는 전자 케이블(도 1에서 요소 (126)으로 도시됨)을 통하여 보안 인터페이스 서버(180)에 의해 전자적으로(electronically) 접속될 수 있는 컴퓨터 상에 호스팅 될 수 있다.

[0055] 메모리(114)는 바람직하게는 다음을 저장한다:

[0056] ● 다양한 기본 시스템 서비스를 조작하고 하드웨어 의존성 작업을 수행하기 위한 프로시저를 포함하는 운영 체제(130);

[0057] ● 보안 인터페이스 서버(180)를 여러 클라이언트 컴퓨터 예컨대 클라이언트 장치(100)(도 1) 및 가능한 경우 또 다른 서버 또는 컴퓨터(예컨대 트랜잭션 서버(200))에 하나 이상의 통신 네트워크, 예컨대 인터넷, 또 다른 광역 네트워크, 근거리 네트워크(예컨대, 로컬 무선 네트워크는 클라이언트 장치(100)를 보안 인터페이스 서버(180)에 연결시킬 수 있음), 도시권 네트워크, 등을 통하여 연결시키기 위하여 사용되는 네트워크 통신 모듈(132);

[0058] ● 클라이언트 컴퓨터로부터 요청을 수신하기 위한 트랜잭션 모듈 서빙 애플리케이션(134);

[0059] ● 사용자 요청 시에, 클라이언트 장치(100)로의 분배를 위한 언브랜디드 트랜잭션 모듈(136)

[0060] ● 보안 인터페이스 서버(180)가 상호대화할 수 있는 컴퓨터/도메인을 특정하는 교차-도메인 정책(138);

[0061] ● 보안 인-애플리케이션 트랜잭션의 기록을 저장하기 위한 트랜잭션 데이터베이스(140/240); 및

[0062] ● 검증된 애플리케이션 자격증명의 데이터베이스(142).

[0063] 보안 인터페이스 서버(180)는 인터넷/네트워크(126)를 통하여 하나 이상의 클라이언트 장치(100)에 연결된다. 도 1은 단지 하나의 이러한 클라이언트 장치(100)에 대한 연결을 나타낸다. 클라이언트 장치(100)가 개인용 컴퓨터(예컨대, 데스크탑 또는 랩탑 컴퓨터)인 것이 가능하며 또는 임의 형태의 모바일 연상 장치(예컨대, 아이폰, 블랙베리, 및 기타) 일 수 있다.

[0064] 전형적인 구체 예에서, 클라이언트 장치(100)는 다음을 포함한다:

[0065] ● 하나 이상의 처리 장치(CPU)(2);

[0066] ● 네트워크 또는 또 다른 통신 인터페이스(10);

[0067] ● 메모리(14);

[0068] ● 선택사항으로, 하나 이상의 선택적인 제어기(18)에 의해 접속 가능한 하나 이상의 자기 디스크 저장 및/또는 영구 저장 장치(20);

[0069] ● 사용자 인터페이스(4), 사용자 인터페이스(4)는 디스플레이(6) 및 키보드 또는 키패드(8)를 포함함;

[0070] ● 전술한 부품들을 상호연결시키기 위한 하나 이상의 통신 버스(12); 및

[0071] ● 전술한 부품들에게 전원 공급하기 위한 전원 공급장치(24), 이러한 전원 공급장치는 예컨대 배터리일 수 있음.

[0072] 일부 구체 예에서, 메모리(14) 내 데이터는 캐싱(caching)과 같은 공지된 연산 기법을 사용하여 선택적인 비-휘발성 메모리(20)와 심리스하게(seamlessly) 공유될 수 있다. 일부 구체 예에서 클라이언트 장치(100)는 자기 디

스크 저장 장치를 갖지 않는다. 예컨대, 일부 구체 예에서, 클라이언트 장치(100)는 휴대용 핸드헬드 연산 장치이며 네트워크 인터페이스(10)는 무선 수단에 의해 인터넷/네트워크(126)과 통신한다.

[0073] 메모리(14)는 바람직하게는 다음을 포함한다:

[0074] ● 다양한 기본 시스템 서비스를 조작하고 하드웨어 의존성 작업을 수행하기 위한 프로시저를 포함하는 운영 체제(30);

[0075] ● 클라이언트 장치(100)를 또 다른 컴퓨터 예컨대 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)에 연결시키기 위해 사용되는 네트워크 통신 모듈(32), 일부 구체 예에서 네트워크 통신 모듈(32)은 선택적인 웹 브라우저, 예컨대 마이크로소프트 인터넷 익스플로러(Microsoft Internet Explorer) 버전 6.0 또는 그 이후 버전, 파이어폭스(Firefox) 2.x, 파이어폭스(Firefox) 3.x, AOL 9, 오페라(Opera) 9.5 또는 그 이후 버전, 사파리(Safari) 3.x, 크롬(Chrome) 2.0 또는 더 높은 버전을 포함하며, 그리고 일부 구체 예에서, 선택적인 웹 브라우저는 플래쉬(FLASH) 플레이어와 같은 모듈을 포함함;

[0076] ● 인-애플리케이션 트랜잭션을 요청할 수 있으며 인-애플리케이션 트랜잭션이 완료되었는지 확인할 수 있는 클라이언트 애플리케이션(36);

[0077] ● 인-애플리케이션 트랜잭션을 초기화하기 위한 요청 모듈(36); 및

[0078] ● 인-애플리케이션 트랜잭션을 수행하는 트랜잭션 모듈(38).

[0079] 트랜잭션 서버(200)는 전형적으로 하나 이상의 처리 장치(CPU)(202), 네트워크 또는 또 다른 통신 인터페이스(210), 메모리(214), 선택적으로 하나 이상의 선택적인 제어기(218)에 의해 접속되는 하나 이상의 자기 디스크 저장 및/또는 비휘발성 장치(220), 전술한 부품들을 상호연결시키기 위한 하나 이상의 통신 버스(212), 및 전술한 부품들에게 전원 공급하기 위한 전원 공급장치(224)를 포함할 것이다. 메모리(214) 내 데이터는 캐싱(caching)과 같은 공지된 연산 기법을 사용하여 비-휘발성 메모리(220)와 심리스하게(seamlessly) 공유될 수 있다. 메모리(214) 및/또는 메모리(220)는 중앙 처리 장치(202)와 관련하여 원격으로 위치한 대용량 저장장치를 포함할 수 있다. 환언하면, 메모리(214) 및/또는 메모리(220)에 저장된 일부 데이터는 실제로 트랜잭션 서버(200) 외부에 있지만 네트워크 인터페이스(210)를 사용하는 인터넷, 인트라넷, 또는 또 다른 형태의 네트워크 또는 전자 케이블(도 1의 요소(126)로 도시됨)을 통하여 트랜잭션 서버(200)에 의해 전자적으로(electronically) 접속될 수 있는 컴퓨터 상에 호스팅 될 수 있다.

[0080] 메모리(214)는 바람직하게는 다음을 저장한다:

[0081] ● 다양한 기본 시스템 서비스를 조작하고 하드웨어 의존성 작업을 수행하기 위한 프로시저를 포함하는 운영 체제(230);

[0082] ● 트랜잭션 서버(200)를 여러 클라이언트 컴퓨터 예컨대 클라이언트 장치(100)(도 1) 및 가능한 경우 또 다른 서버 또는 컴퓨터(예컨대 보안 인터페이스 서버(180))에 하나 이상의 통신 네트워크, 예컨대 인터넷, 또 다른 광역 네트워크, 근거리 네트워크(예컨대, 로컬 무선 네트워크는 클라이언트 장치(100)를 보안 인터페이스 서버(180)에 연결시킬 수 있음), 도시권 네트워크, 등을 통하여 연결시키기 위하여 사용되는 네트워크 통신 모듈(232);

[0083] ● 클라이언트 장치(100) 상에서 작동하는 트랜잭션 모듈(38)로부터의 요청의 검증을 확인하기 위한 트랜잭션 모듈(234);

[0084] ● 트랜잭션 서버(200)가 상호대화할 수 있는 컴퓨터/도메인을 특정하는 교차-도메인 정책(236);

[0085] ● 클라이언트 장치(100) 상에서 수행중인 트랜잭션 모듈(38)로 인-애플리케이션 트랜잭션을 실행하기 위한 애플리케이션 프로그래밍 인터페이스(application programming interface, "API")(238); 및

[0086] ● 보안 인-애플리케이션 트랜잭션의 상태(status)를 저장하기 위한 트랜잭션 데이터베이스(140/240).

[0087] 도 2를 참조하여, 본 발명의 첫 번째 구체 예에 따르는 예시적인 방법이 개시된다. 본 방법은 본 발명에 따라 트랜잭션을 상호대화적으로 서비스하기 위하여, 보안 인터페이스 서버(180), 클라이언트 장치(100), 및 트랜잭션 서버(200)에 의해 수행되는 단계들을 상세하게 제시한다.

[0088] *단계(202).*

[0089] 단계(202)에서, 클라이언트 장치(100)는 로컬 데이터 스토어(예컨대 도 1의 메모리(14) 또는 메모리(20))로부터

의 클라이언트 애플리케이션(34)을 실행하거나 또는 원격 애플리케이션 서버(도시되지 않음)로부터 인터넷 또는 컴퓨터 네트워크를 통하여 클라이언트 애플리케이션(34)을 획득하고 실행시킨다. 일부 경우에, 클라이언트 애플리케이션(34)은 소셜 네트워킹 애플리케이션(예컨대, 페이스북(FACEBOOK), 마이스페이스(MYSPACE)), 재무 서비스 애플리케이션, 회계 애플리케이션, 또는 세무 대리 애플리케이션이다.

[0090] *단계(204)*.

[0091] 클라이언트 애플리케이션(34)이 클라이언트 장치(100) 상에서 실행 중인 동안 일부 시점에서, 클라이언트 애플리케이션(34)은 보안 인-애플리케이션 트랜잭션을 요청하기 위한 요청 모듈(36)을 요구한다. 일부 경우에, 보안 인-애플리케이션 트랜잭션은 클라이언트 애플리케이션의 사용자의 신원과 관련된 계정을 사용하여 인-게임 업그레이드를 구매하기 위한 인-게임 트랜잭션이다. 인-게임 업그레이드의 예는 비제한적으로, 레벨 해제, 가상 장비의 구매, 가상 특수 무기의 구매, 치트(cheat)의 구매, 또는 가상 화폐의 구매를 포함한다.

[0092] 전형적으로, 보안 인-애플리케이션 요청은 애플리케이션 자격증명, 클라이언트 애플리케이션(34) 사용자를 식별하는 정보, 및 고유 트랜잭션 식별자를 포함한다. 애플리케이션 자격증명은 애플리케이션 개발자를 식별하는 클라이언트 애플리케이션(34)과 관련된 자격증명이다. 예컨대, 일부 구체 예에서, 클라이언트 애플리케이션(34)은 게임 애플리케이션이고 애플리케이션 자격증명은 게임 애플리케이션 개발자를 식별한다. 애플리케이션 자격증명은 인-애플리케이션 트랜잭션에 신용처리(credit)할 사람이 누구인지를 결정하기 위해 사용될 수 있다. 예를 들어, 인-애플리케이션 트랜잭션이 클라이언트 애플리케이션(34)의 사용자에 의한 자금의 지불을 포함하는 경우, 애플리케이션 자격증명은 누가 이런 자금에 대해 신용처리 되는지를 식별하기 위해 사용된다. 일반적으로, 애플리케이션 자격증명은 인-애플리케이션 트랜잭션에 대한 제1 당사자(first party)를 식별하기 위해 사용된다. 유사하게, 클라이언트 애플리케이션(34) 사용자를 식별하는 정보는 인-애플리케이션 트랜잭션에 대한 제2 당사자(second party)를 결정하기 위해 사용될 수 있다. 예를 들어, 전형적으로, 제2 당사자는 인-애플리케이션 트랜잭션이 완료되는 경우 이루어질 수 있는 목적(예컨대, 추가 게임 레벨, 더 많은 사용자 특징, 등)을 도모하기 위하여, 클라이언트 애플리케이션(34)에 의해 수행되는 보안 인-애플리케이션 트랜잭션을 실행하는 것을 원하는 클라이언트 애플리케이션(34)의 사용자이다. 고유 트랜잭션 식별자는 인-애플리케이션 트랜잭션의 상태를 추적하기 위하여 사용된다. 전형적인 구체 예에서, 애플리케이션 자격증명, 클라이언트 애플리케이션(34) 사용자를 식별하는 정보, 및 고유 트랜잭션 식별자의 포맷은 보안 인터페이스 서버(180) 및/또는 트랜잭션 서버(200)의 요구조건에 따라 사전 결정된다. 예컨대, 애플리케이션 자격증명은 모든 인-애플리케이션 트랜잭션에서의 사용을 위하여 애플리케이션 개발자에게 제공된 일련 번호(serial number)일 수 있다. 더욱이, 클라이언트 애플리케이션(34) 사용자를 식별하는 정보는, 사용자가 애플리케이션 개발자에 대해 계정을 생성하였을 때 및/또는 사용자가 보안 인터페이스 서버(180) 또는 트랜잭션 서버(200)에 대해 계정을 생성하였을 때 및/또는 애플리케이션 개발자가 사용자를 보안 인터페이스 서버(180) 또는 트랜잭션 서버(200)에 대해 등록하였을 때 생성된, 사용자와 고유하게 관련된 등록 정보일 수 있다. 실시와 무관하게, 클라이언트 애플리케이션(34) 사용자를 식별하는 정보는 보안 인터페이스 서버(180) 및/또는 트랜잭션 서버(200)에 대해 사용자를 고유하게 식별한다. 유사하게, 바람직한 구체 예에서, 고유 트랜잭션 식별자는 클라이언트 애플리케이션(34)의 사용자에 의해 수행되는 단일 인-애플리케이션 트랜잭션을 고유하게 식별한다.

[0093] 따라서, 요컨대, 단계(204) 완료 즉시, 클라이언트 애플리케이션(34) 및 관련된 요청 모듈(36)을 통하여, 클라이언트 애플리케이션(34)이 클라이언트 장치(100) 상에서 실행중인 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청이 발생되며, 여기서 상기 요청은 (i) 클라이언트 애플리케이션에 대한 자격증명, (ii) 클라이언트 애플리케이션의 사용자의 아이디, 및 (iii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자를 포함한다. 보안 인-애플리케이션 트랜잭션에 대한 요청은 인터넷 또는 컴퓨터 네트워크를 통하여 무제한 제1 교차-도메인 정책(138)을 갖는 보안 인터페이스 서버(180)(제1 도메인)에 제출된다.

[0094] *단계(206)*.

[0095] 단계(206)에서, 보안 인-애플리케이션 트랜잭션에 대한 요청은, 보안 인터페이스 서버(180) 상에서 실행중이며, 애플리케이션 사용자를 식별하는 정보, 고유 트랜잭션 식별자, 및 애플리케이션 자격증명을 갖는 트랜잭션 모듈 서빙 애플리케이션(134)에 의해 인터넷 또는 컴퓨터 네트워크를 통하여 수신된다. 일부 구체 예에서, 단계(206)에서 수신된 요청은 사용자의 신원을 포함하지 않는다. 이러한 구체 예에서, 사용자의 신원은 단지 트랜잭션 모듈(38)에 의해 트랜잭션 서버(200)의 트랜잭션 서버 모듈(234)로 통신된다.

[0096] *단계(208)*.

[0097] 단계(208)에서, 클라이언트 애플리케이션(34)에 대한 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 확인된다. 일부 구체 예에서, 검증된 애플리케이션 자격증명의 데이터베이스(142)는 보안 트랜잭션을 수행하기 위해 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)를 사용할 수 있는 각각의 정당한 애플리케이션 개발자의 아이디(identification)를 포함한다.

[0098] *단계(210)*.

[0099] 단계(206)에서 수신된 애플리케이션 요청이 제공된 애플리케이션 자격증명이 확인되지 않으면(210-아니오), 이는 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 존재하지 않거나 또는 데이터베이스가 자격증명이 검증되지 않거나 비활성화된 것을 지시하는 것을 의미하며, 따라서 트랜잭션은 종결된다(212). 도시되지 않은 일부 구체 예에서, 클라이언트 애플리케이션(34)은 이러한 실패를 통보받지 않는다. 도시되지 않은 일부 구체 예에서, 애플리케이션 개발자는 이러한 실패를 통보받지 않는다. 단계(206)에서 수신된 애플리케이션 요청이 제공된 애플리케이션 자격증명이 확인되면(210-예), 이는 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 존재하는 것을 의미하며, 프로세스 제어가 단계(214)로 이동한다.

[0100] *단계(214)*.

[0101] 단계(214)에서, 요청이 트랜잭션 데이터베이스(140/240) 내로 키(key) 된다. 일부 구체 예에서, 이는 개시된 단일 트랜잭션에 대한 트랜잭션 데이터베이스(140/240) 내로 고유 엔트리를 부가하는 것을 포함한다. 요청이 클라이언트 애플리케이션(34)의 사용자의 신원을 포함하는 일부 구체 예에서, 트랜잭션 엔트리가 상기 사용자와 관련된 계정에 부가된다. 트랜잭션은 단계(214)에서 수행되지 않는다. 예를 들어, 돈의 합계는 단계(214) 동안 트랜잭션에서 식별된 사용자에게 신용처리 또는 직불처리 되지 않는다. 이러한 신용처리 또는 직불처리는, 전부 일어나는 경우, 개시된 방법의 후속 단계에서 일어난다. 트랜잭션 데이터베이스(140/240)는 보안 인터페이스 서버(180) 및 트랜잭션 서버(200) 둘 모두에 의해 접근 가능하다. 일부 구체 예에서, 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)는 서로 다른 서버이다. 일부 구체 예에서 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)는 별도의 교차-도메인 정책 및 도메인을 갖는 동일 서버이다.

[0102] *단계(216)*.

[0103] 단계(216)에서, 트랜잭션 모듈 서빙 애플리케이션(134)은 요청된 트랜잭션에 대한 트랜잭션 모듈의 검증을 달성하기 위한 충분한 정보로써 언브랜디드 트랜잭션 모듈(예컨대, SWF 파일 포맷)(136)을 다이내믹하게 브랜딩(brand)하고 브랜디드 트랜잭션 모듈을 클라이언트 장치(100)로 트랜잭션 모듈(38)로서 보낸다. 일부 구체 예에서, 이는 트랜잭션 모듈 서빙 애플리케이션(134)이 하나 이상의 자격증명을 언브랜디드 트랜잭션 모듈(136) 내로 주입시킴으로써, 검증된 트랜잭션 모듈(38)을 다이내믹하게 발생시키도록 하여 달성된다. 이러한 주입(보안) 방법의 예는 예컨대 2009.10.27. 출원된 미국 특허 출원 일련 번호 12/607,005, 명칭 "Systems and Methods for Authenticating an Electronic Transaction"에서 발견되며, 이는 그 전체가 참조로서 본 명세서에 수록된다.

[0104] 일부 구체 예에서, 제2 도메인(180)은 보안 트랜잭션 모듈(38)을 언브랜디드 트랜잭션 모듈(136)의 내용의 변형 없이 클라이언트 장치(100) 상의 클라이언트 애플리케이션(34) 및/또는 요청 모듈(36)에 제공한다. 일부 이러한 구체 예에서, 클라이언트 장치(100)로부터 보안 인터페이스 서버(180)로 보내진 트랜잭션 모듈에 대한 요청은, 트랜잭션 모듈을 검증하는 역할을 하고 언브랜디드 트랜잭션 모듈(136)의 외부에 존재하는 파라미터를 언브랜디드 트랜잭션 모듈(136)에게 제공함으로써 서비스된다. 요청 모듈(36)은 트랜잭션 모듈(136)을 검증할 수 있다(이에 따라 트랜잭션 모듈(136)은 파라미터를 트랜잭션 모듈(136)에 주입하지 않고 트랜잭션 모듈(38)로 간주되는 것이 허용된다). 일부 구체 예에서, 트랜잭션 모듈(38)로 주입되거나 또는 트랜잭션 모듈의 외부에 존재하는 하나 이상의 파라미터로서 제공되는 자격증명은 사용자 식별자 키(key)이다. 일부 구체 예에서, 애플리케이션 사용자 식별자 키에는 클라이언트 장치(100)로부터 유래하고 단계(206)에서 수신된 요청이 제공된다. 일부 구체 예에서, 애플리케이션 사용자 식별자 키는 애플리케이션 개발자에 대하여 사용자가 갖는 계정과 관련되며 이러한 계정은 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)에 의해 서비스된다. 일부 구체 예에서, 애플리케이션 사용자 식별자 키는 제3자, 예컨대 페이스북(FACEBOOK) 또는 마이스페이스(MYSPACE)에 의해 제공된다.

[0105] 일부 구체 예에서, 트랜잭션 모듈(38)로 주입되거나 또는 트랜잭션 모듈의 외부에 존재하는 하나 이상의 파라미터로서 제공되는 자격증명은 기준 시간에 기초하는 솔팅 값(salting value)(146)이다. 일부 구체 예에서, 이러한 솔팅 값(salting value)은 요청과 관련된 협정 세계시(좌표(coordinate)d Universal Time, UTC)이다. 예를

들어, 솔팅 값(146)은 단계(202), 단계(204), 단계(206), 단계(208), 단계(210), 단계(214), 또는 단계(216)의 실시(예컨대, 시작, 종결, 진행 시간) 동안 시간에서 UTC이거나 또는 요청이 클라이언트 장치(100)에 의해 유래되거나 또는 보안 인터페이스 서버(180)에 의해 수신되는 시간의 일부 또 다른 사전결정된 함수일 수 있다. UTC는 지구의 느린 회전을 보상하기 위해 불규칙적인 간격에서 윤초(leap second)가 부가된 국제 원자시(International Atomic Time, TAI)에 기초하는 시간 표준이다. 윤초는 UTC가 그리니치의 왕립 천문대에서의 평균 태양시인 UT1을 근접하게 추적하는 것을 허용하기 위해 사용된다. 일부 구체 예에서, 솔팅 값(146)은 UTC 및 일부 시간 증가의 정수 분할이며, 예컨대 1시간, 8시간, 12시간, 등이다.

[0106] 일부 구체 예에서, 트랜잭션 모듈(38)로 주입되거나 또는 트랜잭션 모듈의 외부에 존재하는 하나 이상의 파라미터로서 제공되는 자격증명은 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)에 의해 공유되는 비밀 키(secret key)이다. 비밀 키(148)의 특징은 인터넷/네트워크(126)를 통하여 통신되지 않으며 단지 애플리케이션 개발자, 트랜잭션 서버(200)의 호스트, 및 보안 인터페이스 서버(180)의 호스트만이 그 신원을 안다는 것이다. 예를 들어 [Section 2.4 of Kaufman, Network Security, Prentice-Hall, Inc., Upper Saddle River, N.J.]를 참조하며, 이는 참조로서 본 명세서에 수록된다.

[0107] 일부 구체 예에서, 트랜잭션 모듈(38)로 주입되거나 또는 트랜잭션 모듈의 외부에 존재하는 하나 이상의 파라미터로서 제공되는 자격증명은 단계(206)에서 수신된 요청이 제공되는 애플리케이션 사용자 식별자이다. 일부 구체 예에서, 전술한 자격증명의 임의 조합은, 이러한 조합이 클라이언트 애플리케이션(34)에 전송되거나 또는 트랜잭션 모듈의 외부에 존재하는 하나 이상의 파라미터로서 제공되기 이전에 트랜잭션 모듈(38)에 주입된다. 일부 구체 예에서, 전술한 자격증명의 임의 조합은 임시 서명 키(signing key)를 발생시키기 위해 사용된다. 예를 들어, 일부 구체 예에서 이러한 자격증명은 함께 절단(truncated)되거나, 또는 또 다르게 조합되고, 그 후 일-방향 해쉬(one-way hashed)되어 서명 키를 발생시키는데 상기 서명 키는, 전술한 자격증명을 주입시키거나 및/또는 트랜잭션 모듈의 외부에 존재하는 이러한 자격증명을 제공하는 것에 대신하여, 또는 이에 부가하여, 트랜잭션 모듈(38) 내에 주입된다. 일부 구체 예에서, 또 다른 자격증명이, 전술한 것에 부가하여, 또는 이를 대신하여, 트랜잭션 모듈(38) 내에 주입된다.

[0108] *단계(218).*

[0109] 단계(218)에서, 클라이언트 애플리케이션(34)은 보안 인터페이스 서버(180)를 자신들의 소스 URL이라고 식별하는 프로그램에 관련되고 이에 한정되는 별도의 도메인-특정 보안 샌드박스에서 트랜잭션 모듈(38)을 실행한다. 예를 들어, 일부 구체 예에서, 클라이언트 애플리케이션(34)은 플래쉬(FLASH) 프로그램이고 트랜잭션 모듈(38)은 단계(218) 동안 클라이언트 애플리케이션(34)에 의해 로딩되고 실행되는 SWF 파일 형태이다. 이러한 구체 예에서, 클라이언트 애플리케이션(34)은 그 소스 URL을 결정하기 위하여 트랜잭션 모듈(38)을 심문한다. 이러한 경우, 트랜잭션 모듈(38)의 소스 URL은 보안 인터페이스 서버(180)의 URL이다. 따라서, 클라이언트 애플리케이션(34)은 보안 인터페이스 서버(180)의 도메인으로부터의 프로그램에 전용되는 도메인-특정 보안 샌드박스 내에 트랜잭션 모듈(38)을 로딩하고 실행한다. 환언하면, 단계(218)에서, 클라이언트 애플리케이션(34)은 검증된 트랜잭션 모듈(38)을 실행하고 이에 따라 검증된 트랜잭션 모듈은 메모리 내 별도의 도메인-특정 보안 샌드박스로 로딩되며, 여기서 (i) 별도의 도메인-특정 보안 샌드박스는 클라이언트 애플리케이션이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되고, (ii) 별도의 도메인-특정 보안 샌드박스는 자신들의 소스 URL을 보안 인터페이스 서버(180)의 도메인이라고 식별하는 프로그램과 관련되고 한정되며, (iii) 검증된 트랜잭션 모듈(38)은, 검증된 트랜잭션 모듈(38)의 소스 URL의 신원이 변하거나 파괴되지 않도록 클라이언트 애플리케이션(34)에 의해 실행되며, 그리고 (iv) 검증된 트랜잭션 모듈(38)은 상기 검증된 트랜잭션 모듈(38)을 내성하는 파워를 클라이언트 애플리케이션(34)에게 허용하지 않는다. 따라서, 유리하게는, 트랜잭션 모듈(38)은 비록 클라이언트 애플리케이션(34)에 의해 로딩되었으나, 클라이언트 장치(100) 상에서 보안 방식으로 실행될 수 있다. 클라이언트 애플리케이션(34)은 트랜잭션 모듈(38)을 내성할 수 없다. 환언하면, 클라이언트 애플리케이션(34)은 트랜잭션 모듈(38)에 의해 저장된 어떠한 값도 판독할 수 없다. 이는 매우 유리한데 왜냐하면 클라이언트 애플리케이션의 사용자가, 클라이언트 애플리케이션(34)이 민감 정보를 획득할 염려 없이, 상기 민감 정보를 트랜잭션 모듈(38)에 기입할 수 있기 때문이다.

[0110] *단계(220).*

[0111] 단계(220)에서, 트랜잭션 모듈(38)은, 모듈(38)이 별도의 도메인-특정 보안 샌드박스에서 실행되는 동안, 트랜잭션 호출을 전송한다. 트랜잭션 호출은 제2 도메인, 즉 트랜잭션 서버(200)의 도메인으로 보내진다. 트랜잭션 서버(200)는 트랜잭션 서버(200)와 트랜잭션 서버(200)(제2 도메인)의 외부에 있는 프로그램 사이의 상호대화를

자신의 소스 URL이 보안 인터페이스 서버(180)(제1 도메인)의 URL인 이러한 외부 프로그램으로 한정하는 제2 교차-도메인 정책(236)을 가진다.

[0112] *단계(222 및 224)*.

[0113] 단계(222)에서, 트랜잭션 모듈(38)을 호출하는 것이 보안 인터페이스 서버(180)를 자신의 소스 URL로서 명명하는 것인지에 대한 결정이 이루어진다. 단계(220)에서 앞서 기술한 바와 같이, 이는 트랜잭션 서버(200)와 상호대화하기 위한 필수적인 조건인데 왜냐하면 트랜잭션 서버(200)의 교차-도메인 정책(236)이 외부 상호대화를 단지 자신의 소스 URL이 보안 인터페이스 서버(180)의 URL인 이러한 프로그램으로 한정하기 때문이다. 트랜잭션 모듈(38)이 보안 인터페이스 서버(180)에 의해 클라이언트 장치(100)로 서비스되고, 그리고 트랜잭션 모듈(38)의 소스 URL이 보존되는 방식으로 클라이언트 애플리케이션(34)에 의해 로딩되기 때문에, 트랜잭션 모듈(38)은 조건(222)을 만족하여야 하며(222-예) 이에 따라 프로세스 제어는 단계(226)로 이동하여야 한다. 트랜잭션 모듈(38)의 소스 URL이 보존되지 않는 방식으로 클라이언트 애플리케이션(34)이 트랜잭션 모듈(38)을 로딩하는 경우(예컨대, 플래쉬(FLASH) 로드바이트(loadByte) 호출 사용에 의함), 조건(222)은 만족되지 않을 것이며(222-아니오) 따라서 프로세스 제어는 단계(224)로 이동하여 여기서 트랜잭션이 실패할 것이다. 이러한 경우, 트랜잭션의 기록은 트랜잭션 데이터베이스(140/240)로부터 삭제될 것이다.

[0114] *단계(226)*.

[0115] 프로세스 제어가 단계(226)에 도달하면, 트랜잭션 서버(200)에서 작동중인 트랜잭션 서버 모듈(234)은 요청된 트랜잭션을 수행하기 위하여 API(238)가 트랜잭션 모듈(38)과 상호대화하는 것을 허용한다. 일부 구체 예에서, 트랜잭션이 시작하기 이전에 추가 보안 수단이 도입된다. 이러한 보안 수단의 예는 2009.10.27. 출원된 미국 특허 출원 일련 번호 12/607,005, 명칭 "Systems and Methods for Authenticating an Electronic Transaction"에 개시되며, 이는 그 전체가 참조로서 본 명세서에 수록된다. 예를 들어, 미국 특허 출원 일련 번호 12/607,005에 개시된 단계(214 내지 218)을 참조하라.

[0116] *단계(228)*.

[0117] 단계(228)에서, 사용자는 트랜잭션 모듈(38) 및 API(238)를 사용하여 트랜잭션을 수행한다. 이러한 트랜잭션 동안, 사용자는 재무 정보 또는 또 다른 형태의 정보 예컨대 신용 카드 정보, 직불 카드 정보, ATM 정보 (예컨대, 개인 식별 번호), 페이팔(PAYPAL) 계정 정보, 자동이체결제소(automatic clearing house, ACH) 이체 정보, 은행 정보, 청구 주소 정보, 우편 주소 정보, 개인 정보 (예컨대, 사회보장번호, 생일, 개인적 질문에 대한 답변, 등), 쿠폰 정보, 환급 정보, 멤버쉽 정보, 정기구독 정보, 로그인 정보, 패스워드 정보, 보안 토큰(예컨대, RSA 챌린지 번호) 등을 기입할 수 있다. 유리하게는, 트랜잭션 모듈(38)이 클라이언트 장치(100) 상의 자신의 도메인-특정 보안 샌드박스 내에서 작동하기 때문에, 클라이언트 애플리케이션(34) 및 요청 모듈(36)은 단계(228)에서 사용자에 의해 기입된 정보를 획득할 수 없다.

[0118] *단계(230)*.

[0119] 단계(228)에서 사용자에 의해 기입된 정보는 API(238)에 의해 처리되어 이에 따라 사용자에게 신용처리 하거나 직불처리 한다. 단계(228 및 230)는 트랜잭션을 완료하기 위하여 여러 번 반복될 수 있음이 이해될 것이다. 예를 들어, API(238)는 사용자로부터의 하나 이상의 챌린지 또는 요청을 보낼 수 있다. 트랜잭션 모듈(38)을 사용하는 사용자는 이러한 정보를 기입한다. 사용자가 올바른 챌린지 정보를 기입한 경우, API(238)는 사용자로부터 더 많은 정보를 요청함으로써 트랜잭션의 후속 단계로 진행한다. 또한, 트랜잭션의 상태가 트랜잭션에 할당된 고유 트랜잭션 식별자를 사용하는 트랜잭션 데이터베이스(140/240)에 저장된다. 전형적인 구체 예에서, 트랜잭션 서버(200) 및 보안 인터페이스 서버(180) 둘 모두가 트랜잭션 데이터베이스(140/240)에 대하여 접속한다는 점에 주목하라. 보안 인터페이스 서버(180)는 트랜잭션에 할당된 고유 트랜잭션 식별자를 사용하여 단계(214)에서 트랜잭션을 위한 엔트리를 생성하며 한편 트랜잭션 서버(200)는 단계(230) 동안 트랜잭션의 상태(예컨대, 완료, 실패, 금액 신용처리 됨, 금액 직불처리 됨 등)를 기록한다.

[0120] *단계(232-234)*.

[0121] 바람직한 구체 예에서, API(238)의 예가 단일 트랜잭션을 위하여 사용된다. 따라서, 이러한 바람직한 구체 예에서, 트랜잭션을 촉진하기 위하여 사용되었던 API(238)의 예가 종료된다. 또 다른 구체 예에서, API(238)는 다중 트랜잭션 전반에서 영구적이다. 어떠한 형태의 구체 예가 사용되는가와 무관하게, 트랜잭션 서버 모듈(234)은 단계(226 및 230)에서 상호대화하였던 트랜잭션 모듈(38)의 사건을 검증된 것으로 간주하는 것을 중지한다. 이는 유리하게는 개시된 시스템 및 방법의 보안을 강화시키는데 왜냐하면, 나쁜 경우의 양상에서, 트랜잭션 모듈

(38)이 단지 단일의 고유한 트랜잭션을 위하여 단일 사용자에 의해 사용될 수 있기 때문이다. 단계(234)에서, 트랜잭션 모듈(38)이 종료된다.

[0122] *단계(236-242).*

[0123] 앞서 기재한 바와 같이, 트랜잭션 모듈(38)은 클라이언트 장치(100) 상의 자신의 도메인-특정 보안 샌드박스에서 작동하며 트랜잭션 모듈(38)은 내성하기 위한 파워를 클라이언트 애플리케이션(34)에게 허용하지 않는다. 따라서, 클라이언트 애플리케이션(34)은 인-애플리케이션 트랜잭션이 완료되었는지 여부, 및 더욱이 인-애플리케이션 트랜잭션이 성공적으로 완료되어 이에 따라 사용자에게 인-애플리케이션 트랜잭션과 관련된 모든 이익(예컨대, 더 많은 게임 포인트, 등)이 허여되는지 여부를 트랜잭션 모듈(38)로부터 직접적으로 확인하지 않는다. 따라서, 전술한 단계의 일부 또는 전부와 동시에, 또는 전술한 단계의 모두가 완료된 이후에, 클라이언트 애플리케이션(34)은 트랜잭션의 상태를 결정하기 위하여 보안 인터페이스 서버(180)를 사용하는 트랜잭션 데이터베이스(140/204)를 폴링한다. 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)가 트랜잭션 데이터베이스(140/240)에 대하여 접근하는 한편, 보안 인터페이스 서버(180)는 허용성 교차-도메인 정책(138)을 가지며 반면 트랜잭션 서버(200)는 제한 교차-도메인 정책(236)을 가진다는 것이 이해될 것이다. 따라서, 클라이언트 애플리케이션(34)의 소스 URL이 보안 인터페이스 서버(180)의 URL이 아닌 경우, 클라이언트 애플리케이션(34)은 트랜잭션 서버(200)와 직접적으로 상호대화할 수 없을 것이다. 따라서, 클라이언트 애플리케이션(34)은 트랜잭션 서버(200)를 사용하는 트랜잭션 데이터베이스(140/240)를 폴링(poll)하지 못할 것이다.

[0124] 전형적인 구체 예에서, 클라이언트 애플리케이션(34)은 트랜잭션 데이터베이스(140/240)가 트랜잭션이 완료되었는지 여부를 지시할 때까지, 주기적으로(예컨대, 매 분, 매 5분, 매 30분, 등) 보안 인터페이스 서버(180) 상의 트랜잭션 데이터베이스(140/240)를 폴링한다. 전형적인 구체 예에서, 이러한 질의는 매우 제한적이다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34)은 고유 트랜잭션 식별자 및 애플리케이션 자격증명을 제공한다. 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 검증된 경우, 클라이언트 애플리케이션(34)에 의해 제공된 트랜잭션 식별자와 고유하게 관련된 트랜잭션의 상태가 클라이언트에서 실행중인 클라이언트 애플리케이션(34)으로 인터넷 또는 컴퓨터 네트워크를 통하여 되돌려 보내진다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34)은 트랜잭션 데이터베이스(140/240)로부터 개인 정보를 획득하는 것이 허용되지 않는다.

[0125] 첫 번째 구체 예의 실시예와 관련한 처리 단계의 상세사항이 개시된 지금, 본 출원의 장점이 강조될 수 있다. 보안 트랜잭션은 클라이언트 애플리케이션을 사용하여 수행될 수 있는데 여기서 클라이언트 애플리케이션은 트랜잭션을 수행하기 위해 필요한 보안 정보의 지식을 획득할 수 없다. 그럼에도, 트랜잭션을 위한 고유 트랜잭션 식별자를 사용하여, 클라이언트 애플리케이션은 트랜잭션이 성공적이었는지 여부를 결정할 수 있다. 따라서, 개시된 시스템을 사용하여, 클라이언트 애플리케이션은 개시된 보안 플랫폼(예컨대, 보안 인터넷 서버(180) 및 트랜잭션 서버(200))을 설정할 필요가 없는 보안 방식으로 인-애플리케이션 트랜잭션을 지원할 수 있다. 예를 들어, 보안 플랫폼은 별도의 기구에 의해 실행될 수 있다. 이는 클라이언트 애플리케이션을 개발하기 위한 비용을 감소시킨다.

[0126] *인-트랜잭션 사용자 인터페이스.*

[0127] 일부 구체 예에서, 인-애플리케이션 트랜잭션의 룩앤필(look and feel)은 트랜잭션 모듈(38)이 클라이언트 애플리케이션(34)에 의해 제어되는 외형을 생성함으로써 강화된다. 일부 구체 예에서, 이는 요청 모듈(36)과 관련된 애플리케이션 프로그래밍 인터페이스를 통하여 이루어진다. 클라이언트 애플리케이션(34)은 트랜잭션 모듈(38)을 개방(open up)하기 위하여 클라이언트 애플리케이션(34)에 의해 사용되는 스크린 실제 영역의 일부를 할당하기 위하여 애플리케이션 프로그래밍 인터페이스를 사용한다. 한 실시예에서, 트랜잭션 모듈(38)은 300 x 400 픽셀 인터페이스이며, 클라이언트 애플리케이션(34)은 트랜잭션 요청을 생성할 때 이러한 300 x 400 픽셀 인터페이스가 개방하는 스크린상의 좌표(coordinate)를 특정한다. 따라서, 유리하게는, 이러한 구체 예에서, 비록 트랜잭션 모듈(38)이 자신의 도메인-특정 보안 샌드박스에서 실행 중이지만, 클라이언트 애플리케이션(34) 및 트랜잭션 모듈(38)은 동일한 애플리케이션에 나타난다. 일부 구체 예에서, 요청 모듈(36)과 관련된 API 인터페이스는 트랜잭션 모듈(38)에서 사용되는 폰트 및 색상을 특정하기 위한 파라미터를 더욱 포함한다. 이러한 파라미터는 더욱이 클라이언트 애플리케이션 및 트랜잭션 모듈(38)이 동일 애플리케이션에서 나타나도록 한다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34)은 자신의 윈도우에서 실행되며 요청 모듈(36)의 API를 통하여 트랜잭션 모듈(38)에 제공된 모든 픽셀 좌표는 상기 윈도우를 기준으로 한다.

[0128] *보안 인터페이스 서버(180)/트랜잭션 서버(200).*

[0129] 전형적인 구체 예에서, 도 1에 도시된 바와 같이, 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)는 각각 별도의 물리적 서버이다. 일부 대안적인 구체 예에서, 보안 인터페이스 서버(180) 및 트랜잭션 서버(200)는 동일한 물리적 서버에 의해 호스팅 된다. 이러한 구체 예에서, 이러한 물리적 서버는 인터넷 또는 컴퓨터 네트워크(126)를 통하여 클라이언트 장치(100)에 접근할 수 있다.

[0130] *두 번째 구체 예.*

[0131] 본 발명의 두 번째 구체 예가 도 3 및 4에 제시된다. 더욱 상세하게는, 본 발명의 두 번째 구체 예에 따르는 시스템이 도 3과 결합되어 기재된다. 이에 따라, 도 3은 본 발명에 따르는 환경의 토폴로지를 나타낸다. 토폴로지에, 보안 인터페이스 서버(180B), 클라이언트 장치(100B), 트랜잭션 서버(200), 및 요청 모듈 서버(300)가 존재한다. 물론, 또 다른 토폴로지가 가능한데, 예컨대, 보안 인터페이스 서버(180B)는 실제로 수 개의 서버를 포함할 수 있다. 더욱이, 전형적으로, 수백, 수천, 수십만 개 또는 그 이상의 클라이언트 장치(100B)가 존재한다. 도 3에 제시된 예시적인 토폴로지는 단지 해당 분야의 통상의 기술자에게 용이하게 이해되고자 하는 방식으로 본 발명의 두 번째 구체 예의 특징을 설명하는 역할을 한다.

[0132] 도 3A의 클라이언트 장치(100B), 및 클라이언트 장치(100B)에 의해 호스팅 되는 모듈 각각은 도 1A의 클라이언트 장치(100) 내 이들의 유사-명칭 대응부품과 동일하지만 다만, 클라이언트 장치(100B)의 클라이언트 애플리케이션(34B)이 클라이언트 장치(100B)에 의해 요구되는 시간에, 클라이언트 애플리케이션(34B)이 요청 모듈(36)의 전체를 포함하지 않는다는 점이 다르다.

[0133] 도 3A의 트랜잭션 서버(200), 및 도 3A의 트랜잭션 서버(200)에 의해 호스팅 되는 모듈 각각은 도 1A의 트랜잭션 서버(200) 내 이들의 유사-명칭 대응부품과 동일하다.

[0134] 도 3B의 보안 인터페이스 서버(180B), 및 보안 인터페이스 서버(180B)에 의해 호스팅 되는 모듈 각각은 도 1A의 보안 인터페이스 서버(180) 내 이들의 유사-명칭 대응부품과 동일하지만 다만, 보안 인터페이스 서버(180B)의 교차-도메인 정책(138B)(도 3B)가 보안 인터페이스 서버(180B)와 자신의 소스 URL이 요청 모듈 서버(300)의 도메인인 이러한 프로그램 사이의 상호대화를 제한하는 점이 다르다.

[0135] 도 3B에 도시된 바와 같이, 요청 모듈 서버(300)은 전형적으로 하나 이상의 처리 장치(CPU)(302), 네트워크 또는 또 다른 통신 인터페이스(310), 메모리(314), 선택적으로 하나 이상의 제어기(318)에 의해 접근되는 하나 이상의 자기 디스크 저장 및/또는 영구 장치(320), 전술한 부품들을 상호연결시키기 위한 하나 이상의 통신 버스(312), 및 전술한 부품들에게 전원 공급하기 위한 전원 공급장치(324)를 포함할 것이다. 메모리(314) 내 데이터는 캐싱(caching)과 같은 공지된 연산 기법을 사용하여 비-휘발성 메모리(320)와 심리스하게(seamlessly) 공유될 수 있다. 메모리(314) 및/또는 메모리(320)는 중앙 처리 장치(302)와 관련하여 원격으로 위치한 대용량 저장 장치를 포함할 수 있다. 환언하면, 메모리(314) 및/또는 메모리(320)에 저장된 일부 데이터는 실제로 요청 모듈 서버(300) 외부에 있지만 네트워크 인터페이스(310)를 사용하는 인터넷, 인트라넷, 또는 또 다른 형태의 네트워크 또는 전자 케이블(도 3B에서 요소(126)로서 제시됨)을 통하여 요청 모듈 서버(300)에 의해 전자적으로(electronically) 접속될 수 있는 컴퓨터 상에 호스팅 될 수 있다.

[0136] 메모리(314)는 바람직하게는 다음을 저장한다:

[0137] ● 다양한 기본 시스템 서비스를 조작하고 하드웨어 의존성 작업을 수행하기 위한 프로시저를 포함하는 운영 체제(330);

[0138] ● 요청 모듈 서버(300)를 여러 클라이언트 컴퓨터 예컨대 클라이언트 장치(100)(도 3A) 및 가능한 경우 또 다른 서버 또는 컴퓨터(예컨대 트랜잭션 서버(200) 및 보안 인터페이스 서버(180))에 하나 이상의 통신 네트워크, 예컨대 인터넷, 또 다른 광역 네트워크, 근거리 네트워크(예컨대, 로컬 무선 네트워크는 클라이언트 장치(100)를 요청 모듈 서버(300)에 연결시킬 수 있음), 도시권 네트워크, 등을 통하여 연결시키기 위하여 사용되는 네트워크 통신 모듈(332);

[0139] ● 클라이언트 장치(100)로부터 요청을 수신하고 이에 대한 응답으로 요청 모듈(36)을 제공한 트랜잭션 요청 모듈 서빙 애플리케이션(334);

[0140] ● 요청 모듈 서버(300)가 상호대화할 수 있는 컴퓨터/도메인을 특정하는 교차-도메인 정책(336);

[0141] ● 보안 인-애플리케이션 트랜잭션의 기록을 저장하기 위한 트랜잭션 데이터베이스(140/240); 및

[0142] ● 검증된 애플리케이션 자격증명의 데이터베이스(142).

[0143] 도 4를 참조하여, 본 발명의 두 번째 구체 예에 따르는 예시적인 방법이 개시된다. 본 방법은 본 발명의 두 번째 구체 예에 따라 트랜잭션을 상호대화적으로 서비스하기 위하여, 보안 인터페이스 서버(180B), 클라이언트 장치(100), 트랜잭션 서버(200), 및 요청 모듈 서버(300)에 의해 수행되는 단계들을 상세하게 제시한다.

[0144] *단계(402-406).*

[0145] 도 4(구체 예 2)에 개시된 방법은 도 2(구체 예 1)의 방법 및 클라이언트 애플리케이션(34B)이 요구될 때 클라이언트 애플리케이션(34B)이 요청 모듈 서버(300)로부터 요청 모듈(36)을 획득하도록 요구하는 추가 단계를 포함한다. 도 1에 개시된 구체 예에서, 클라이언트 애플리케이션(34B)은 이미 이러한 요청 모듈(36)을 복제하였다. 클라이언트 애플리케이션(34/34B)이 플래쉬(FLASH) 애플리케이션인 일부 구체 예에서, 요청 모듈(36)은 구체 예 1(도 1)에서 컴파일(compile) 된 SWC 파일인 반면, 구체 예 2에서, 요청 모듈(36)은 SWF 파일이다.

[0146] 단계(402)에서, 클라이언트 장치(100B)는 로컬 데이터 스토어로부터 또는 원격 애플리케이션 서버로부터 클라이언트 애플리케이션(34B)을 요구한다. 클라이언트 애플리케이션(34B)이 요구되는 시간에, 클라이언트 애플리케이션(34B)은 인-애플리케이션 보안 트랜잭션을 촉진하기 위한 요청 모듈(36)을 포함하지 않는다. 단계(404)에서, 클라이언트 애플리케이션(34B)은 요청 모듈 서버(300)로부터 요청 모듈(36)을 요구함으로써 트랜잭션을 시작한다. 단계(406)에서, 단계(404)의 요청에 응답하여, 트랜잭션 요청 모듈 서빙 애플리케이션(334)은 요청을 처리하고 요청 모듈(36)을 클라이언트 장치(100)로 보낸다. 단계(406)의 일부 구체 예에서, 클라이언트 애플리케이션(34B)이 트랜잭션 요청 모듈 서빙 애플리케이션(334)이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 확인하는 애플리케이션 자격증명을 제공하는 경우, 트랜잭션 요청 모듈 서빙 애플리케이션(334)은 단지 요청 모듈(36)을 클라이언트 애플리케이션(34B)에 제공한다.

[0147] *단계(408).*

[0148] 단계(408)에서, 클라이언트 애플리케이션(34B)은 요청 모듈 서버(300)를 자신들의 소스 URL이라고 식별하는 프로그램에 관련되고 이에 한정되는 별도의 도메인-특정 보안 샌드박스(제1 샌드박스)에서 요청 모듈(36)을 실행한다. 예를 들어, 일부 구체 예에서, 클라이언트 애플리케이션(34B)은 플래쉬(FLASH) 프로그램이고 요청 모듈(36)은 단계(408) 동안 클라이언트 애플리케이션(34B)에 의해 로딩되고 실행되는 SWF 파일 형태이다. 클라이언트 애플리케이션(34B)은 그 소스 URL을 결정하기 위하여 요청 모듈(36)을 심문한다. 이러한 경우, 요청 모듈(36)의 소스 URL은 요청 모듈 서버(300)의 URL이다. 따라서, 클라이언트 애플리케이션(34B)은 요청 모듈 서버(300)의 도메인으로부터의 프로그램에 전용되는 도메인-특정 보안 샌드박스 내에 요청 모듈(36)을 로딩하고 실행한다. 환언하면, 단계(408)에서, 클라이언트 애플리케이션(34B)은 요청 모듈(36)을 실행하고 이에 따라 요청 모듈(36)은 메모리 내 별도의 도메인-특정 보안 샌드박스로 로딩되며, 여기서 (i) 별도의 도메인-특정 보안 샌드박스는 클라이언트 애플리케이션(34B)이 실행되는 상기 메모리 내 메모리 스페이스로부터 격리되고, (ii) 별도의 도메인-특정 보안 샌드박스는 자신들의 소스 URL을 요청 모듈 서버(300)의 도메인이라고 식별하는 프로그램과 관련되고 한정되며, (iii) 요청 모듈(36)은, 요청 모듈(36)의 소스 URL의 신원이 변하거나 파괴되지 않도록 클라이언트 애플리케이션(34B)에 의해 실행되며, 그리고 (iv) 요청 모듈은 상기 요청 모듈(36)을 내성하는 파워를 클라이언트 애플리케이션(34B)에게 허용하지 않는다. 따라서, 유리하게는, 요청 모듈(36)은 비록 클라이언트 애플리케이션(34B)에 의해 로딩되었으나, 클라이언트 장치(100B) 상에서 보안 방식으로 실행될 수 있다. 클라이언트 애플리케이션(34B)은 요청 모듈(36)을 내성할 수 없다. 환언하면, 클라이언트 애플리케이션(34B)은 요청 모듈(36)에 의해 저장된 어떠한 값도 판독할 수 없다.

[0149] *단계(410).*

[0150] 클라이언트 애플리케이션(34B)이 클라이언트 장치(100B) 상에서 실행 중인 일부 시점에서, 클라이언트 애플리케이션(34B)은 보안 인-애플리케이션 트랜잭션을 요청하기 위한 요청 모듈(36)을 요구한다. 전형적으로, 보안 인-애플리케이션 요청은 애플리케이션 자격증명, 클라이언트 애플리케이션(34B) 사용자를 식별하는 정보, 및 고유 트랜잭션 식별자를 포함한다. 전형적인 구체 예에서, 애플리케이션 자격증명, 클라이언트 애플리케이션(34B) 사용자를 식별하는 정보, 및 고유 트랜잭션 식별자의 포맷은 보안 인터페이스 서버(180B) 및/또는 트랜잭션 서버(200)의 요구조건에 따라 사전 결정된다. 실시와 무관하게, 클라이언트 애플리케이션(34B) 사용자를 식별하는 정보는 보안 인터페이스 서버(180B) 및/또는 트랜잭션 서버(200) 및/또는 요청 모듈 서버(300)에 대해 사용자를 고유하게 식별한다. 유사하게, 바람직한 구체 예에서, 고유 트랜잭션 식별자는 클라이언트 애플리케이션(34B)의 사용자에 의해 수행되는 단일 인-애플리케이션 트랜잭션을 고유하게 식별한다.

[0151] 따라서, 요컨대, 단계(410) 완료 즉시, 클라이언트 애플리케이션(34B) 및 관련된 요청 모듈(36)을 통하여, 클라이언트 애플리케이션(34B)이 클라이언트 장치(100B) 상에서 실행중인 시간에, 보안 인-애플리케이션 트랜잭션과 관련된 요청이 발생되며, 여기서 상기 요청은 (i) 클라이언트 애플리케이션(34B)에 대한 자격증명, (ii) 클라이언트 애플리케이션(34B)의 사용자의 아이디, 및 (iii) 상기 요청을 고유하게 식별하는 트랜잭션 식별자를 포함한다. 보안 인-애플리케이션 트랜잭션에 대한 요청은 인터넷 또는 컴퓨터 네트워크를 통하여 보안 인터페이스 서버(180B)에 제출된다. 여기서, 첫 번째 구체 예와는 달리, 보안 인터페이스 서버(180B)는, 요청 모듈 서버(300)의 도메인인 소스 URL을 갖기 위하여 보안 인터페이스 서버(180B)와 상호대화하는 애플리케이션을 요구하는 제한 교차-도메인 정책(138B)을 가진다. 일부 구체 예에서, 특히 애플리케이션 자격증명이 단계(404)에서 요청 모듈 서버(300)로 이미 제공된 경우, 단계(410)에서 전송되고 단계(412)에서 처리된 요청은 애플리케이션 자격증명을 갖지 않을 수도 있다. 일부 구체 예에서, 단계(410)에서 전송되고 단계(412)에서 처리된 요청은 클라이언트 애플리케이션(34B)의 사용자의 신원을 갖지 않을 수도 있다.

[0152] *단계(412).*

[0153] 단계(410)와 관련하여 전술한 바와 같이, 보안 인-애플리케이션 트랜잭션에 대한 요청은, 보안 인터페이스 서버(180B) 상에서 실행 중이며, 애플리케이션 사용자를 식별하는 정보(선택사항), 고유 트랜잭션 식별자, 및 애플리케이션 자격증명(선택사항)을 갖는 트랜잭션 모듈 서빙 애플리케이션(134)에 의해 인터넷 또는 컴퓨터 네트워크를 통하여 수신된다. 일부 구체 예에서, 단계(412)에서 수신된 요청은 사용자의 신원을 포함하지 않는다. 이러한 구체 예에서, 사용자의 신원은 단지 트랜잭션 모듈(38B)에 의해 트랜잭션 서버(200)의 트랜잭션 서버 모듈(234)로 통신된다.

[0154] *단계(414).*

[0155] 단계(414)의 일부 구체 예에서, 클라이언트 애플리케이션(34B)에 대한 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 확인된다. 일부 구체 예에서, 이러한 확인은 단계(406)에서 이미 수행되었다. 일부 구체 예에서, 이러한 확인은 단계(414 및 406)에서 수행된다. 일부 구체 예에서, 검증된 애플리케이션 자격증명의 데이터베이스(142)는 보안 트랜잭션을 수행하기 위해 보안 인터페이스 서버(180B) 및 트랜잭션 서버(200)를 사용할 수 있는 각각의 정당한 애플리케이션 개발자의 아이디(identification)를 포함한다. 단계(414)에서, 요청 모듈(36)의 소스 URL이 보안 인터페이스 서버(180B)의 교차 도메인 정책(138B)에 대하여 확인되는지 여부를 확인하기 위하여 추가 체크가 수행된다.

[0156] *단계(416).*

[0157] 단계(412)에서 수신된 애플리케이션 요청이 제공된 애플리케이션 자격증명이 확인되지 않거나 및/또는 요청 모듈(36)의 소스 URL이 요청 모듈 서버(300)의 도메인의 URL이 아닌 경우 (416-아니오), 트랜잭션은 실패로 종결된다(418). 도시되지 않은 일부 구체 예에서, 클라이언트 애플리케이션(34B)은 이러한 실패를 통보 받는다. 도시되지 않은 일부 구체 예에서, 애플리케이션 개발자는 이러한 실패를 통보 받는다. 단계(412)에서 수신된 애플리케이션 요청이 제공된 애플리케이션 자격증명이 확인되고, 요청 애플리케이션(36)의 소스 URL이 요청 모듈 서버(300)의 도메인의 URL인 경우(416-예), 프로세스 제어는 도 2A의 단계(214)로 이동한다. 환언하면, 두 번째 구체 예는 첫 번째 구체 예의 단계(214 내지 234)를 포함한다.

[0158] *단계(460 내지 466).*

[0159] 트랜잭션 모듈(38)은 클라이언트 장치(100B) 상의 자신의 도메인-특정 보안 샌드박스에서 작동하며 트랜잭션 모듈(38)은 내성하기 위한 파워를 클라이언트 애플리케이션(34B)에게 허용하지 않는다. 따라서, 클라이언트 애플리케이션(34)B은 인-애플리케이션 트랜잭션이 완료되었는지 여부, 및 더욱이 인-애플리케이션 트랜잭션이 성공적으로 완료되어 이에 따라 사용자에게 인-애플리케이션 트랜잭션과 관련된 모든 이익(예컨대, 더 많은 게임 포인트, 등)이 허여되는지 여부를 트랜잭션 모듈(38)로부터 직접적으로 확인하지 않는다. 따라서, 전술한 단계의 일부 또는 전부와 동시에, 또는 전술한 단계의 모두가 완료된 이후에, 클라이언트 애플리케이션(34B)은 트랜잭션의 상태를 결정하기 위하여 요청 모듈 서버(300)를 사용하는 트랜잭션 데이터베이스(140/204)를 폴링한다.

[0160] 보안 인터페이스 서버(180B), 트랜잭션 서버(200), 및 요청 모듈 서버(300) 각각이 트랜잭션 데이터베이스(140/240)에 대하여 접근하는 한편, 요청 모듈 서버(300)은 구체 예 2의 허용성 교차-도메인 정책(138)을 가지며 반면 트랜잭션 서버(200)는 제한 교차-도메인 정책(236)을 가지며 그리고 보안 인터페이스 서버(180B)는 제한 교차-도메인 정책(138B)을 가진다는 것이 이해될 것이다. 따라서, 클라이언트 애플리케이션(34)의 소스 URL이 보안 인터페이스 서버(180B)의 URL이 아닌 경우, 클라이언트 애플리케이션(34B)은 트랜잭션 서버(200)와 직

접적으로 상호대화할 수 없을 것이다. 더욱이, 클라이언트 애플리케이션(34)의 소스 URL이 요청 모듈 서버(300)의 URL이 아닌 경우, 클라이언트 애플리케이션(34B)은 보안 인터페이스 서버(180B)와 직접적으로 상호대화할 수 없을 것이다. 따라서, 이러한 경우, 클라이언트 애플리케이션(34B)은 트랜잭션 서버(200) 또는 보안 인터페이스 서버(180B)를 사용하는 트랜잭션 데이터베이스(140/240)를 폴링하지 못할 것이다.

[0161] 전형적인 구체 예에서, 클라이언트 애플리케이션(34B)은 트랜잭션 데이터베이스(140/240)가 트랜잭션이 완료되었다고 지시할 때까지, 주기적으로(예컨대, 매 분, 매 5분, 매 30분, 등) 요청 모듈 서버(300) 상의 트랜잭션 데이터베이스(140/240)를 폴링한다. 전형적인 구체 예에서, 이러한 질의는 매우 제한적이다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34B)은 고유 트랜잭션 식별자 및 애플리케이션 자격증명을 제공한다. 애플리케이션 자격증명이 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 검증된 경우, 클라이언트 애플리케이션(34B)에 의해 제공된 트랜잭션 식별자와 고유하게 관련된 트랜잭션의 상태가 클라이언트 장치(100B) 상에서 실행중인 클라이언트 애플리케이션(34B)으로 인터넷 또는 컴퓨터 네트워크를 통하여 되돌려 보내진다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34)은 트랜잭션 데이터베이스(140/240)로부터 개인 정보를 획득하는 것이 허용되지 않는다. 전형적인 구체 예에서, 클라이언트 애플리케이션(34B)은 트랜잭션 데이터베이스(140/240)으로부터 개인 정보를 획득하는 것이 허용되지 않는다.

[0162] **인용된 참조 및 대안적인 구체 예**

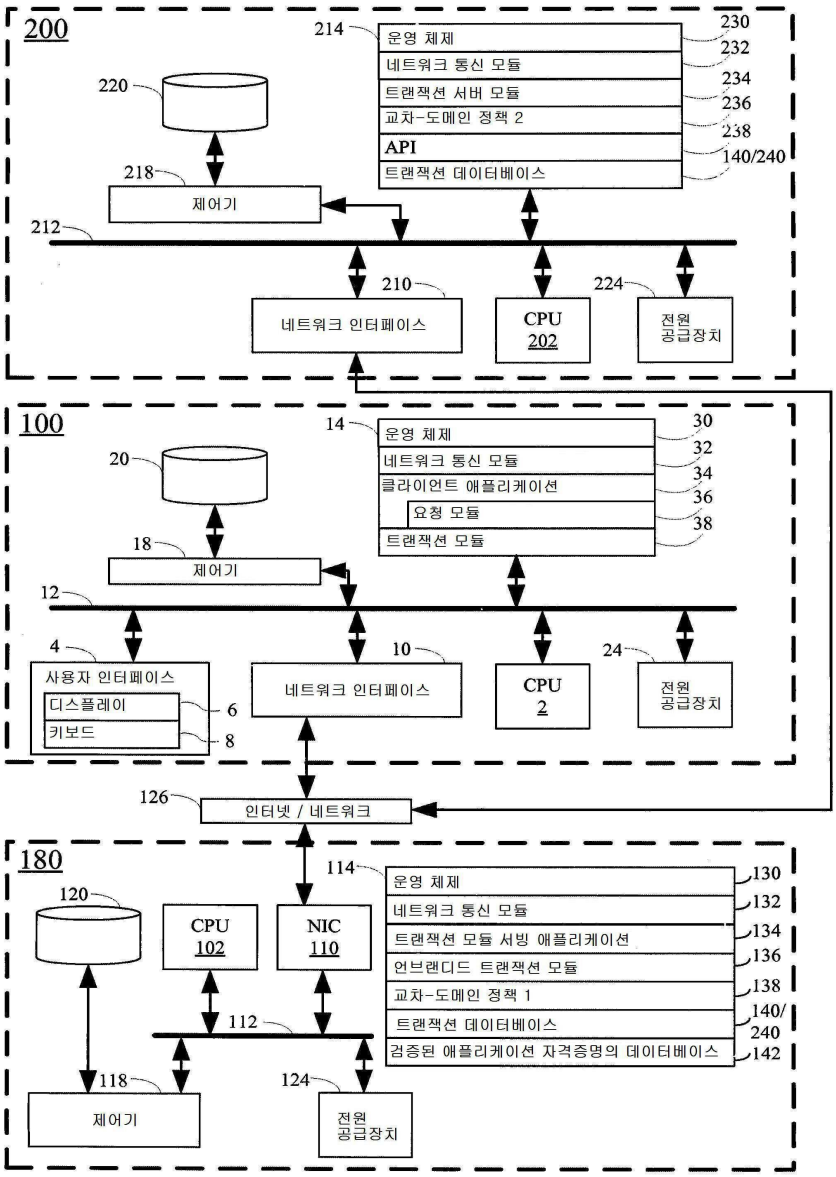
[0163] 본 명세서에서 인용된 모든 참조는, 각 개별 공보 또는 특허 또는 특허 출원이 모든 목적을 위하여 그 전체가 참고문헌으로 수록되는 것으로 구체적이고 개별적으로 지시된 것과 동일한 정도에서 모든 목적으로 위하여 그 전체가 참고문헌으로서 본 명세서에 수록된다.

[0164] 본 발명은 컴퓨터 판독 가능한 저장 매체에 수록된 컴퓨터 프로그램 메커니즘을 포함하는 컴퓨터 프로그램 제품으로서 실시될 수 있다. 예컨대, 컴퓨터 프로그램 제품은 도 1 및/또는 도 3에 제시된 프로그램 모듈을 포함할 수 있다. 이러한 프로그램 모듈은 CD-ROM, DVD, 자기 디스크 저장 제품, 또는 또 다른 유형의 컴퓨터 판독 가능한 데이터 또는 프로그램 저장 제품에 저장될 수 있다.

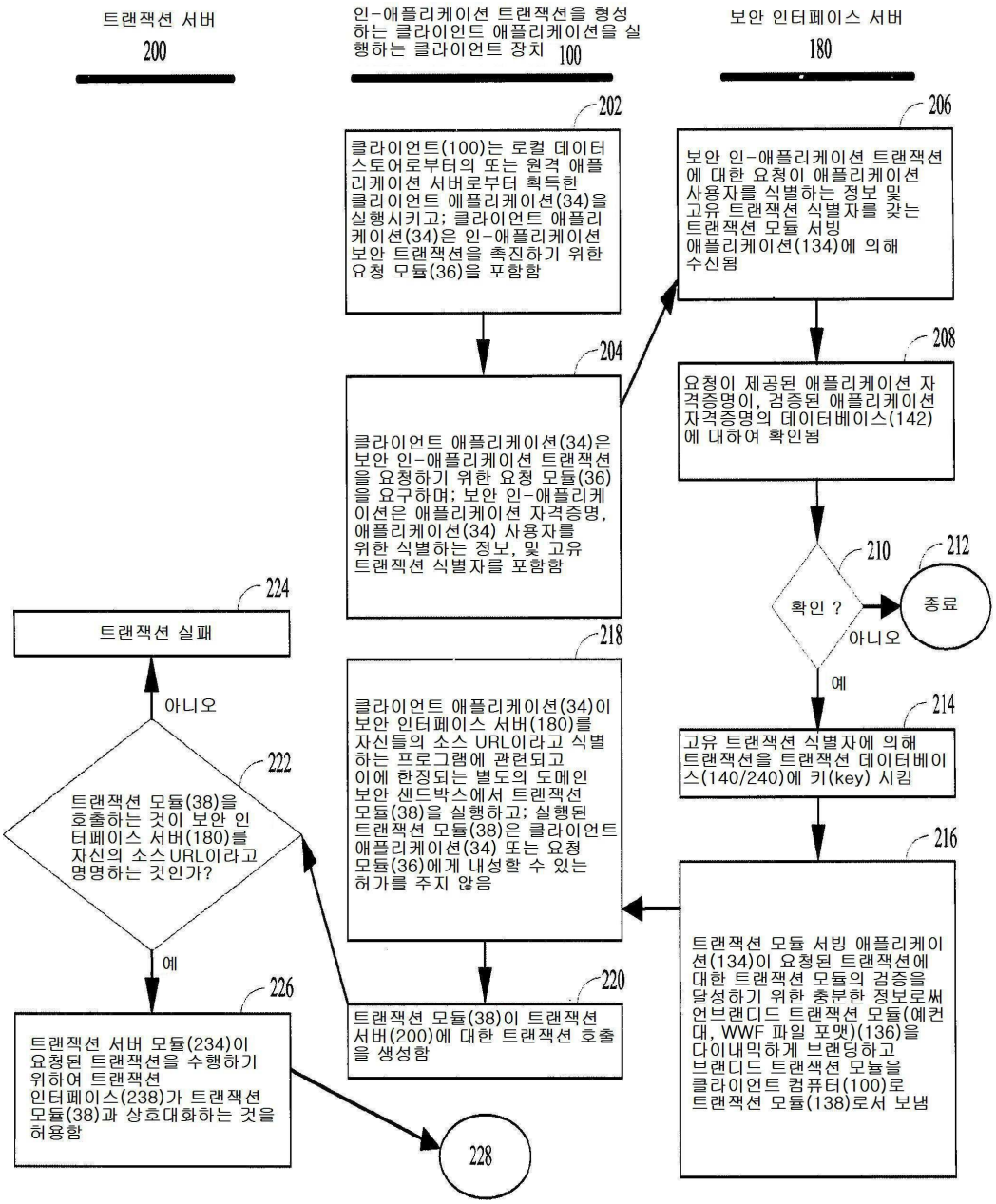
[0165] 본 발명의 많은 변형 및 변화가 본 발명의 사상 및 범위를 벗어나지 않으면서 이루어질 수 있으며, 이는 해당 분야의 통상의 기술자에게 명백할 것이다. 본 명세서에 기재된 특정 구체 예는 단지 예시를 위하여 제공된다. 구체 예는 본 발명의 원칙 및 그 실제적 적용을 최선으로 설명하고, 이에 따라 해당 분야의 또 다른 통상의 기술자가, 고려된 특정 용도에 적합한 다양한 변형을 갖는 다양한 구체 예 및 발명을 가장 잘 이용하도록 하기 위하여 선택되고 기재되었다. 본 발명은 단지 첨부된 청구항, 및 이러한 청구항에 부여된 완전한 균등 범위에 의해서만 제한될 것이다.

도면

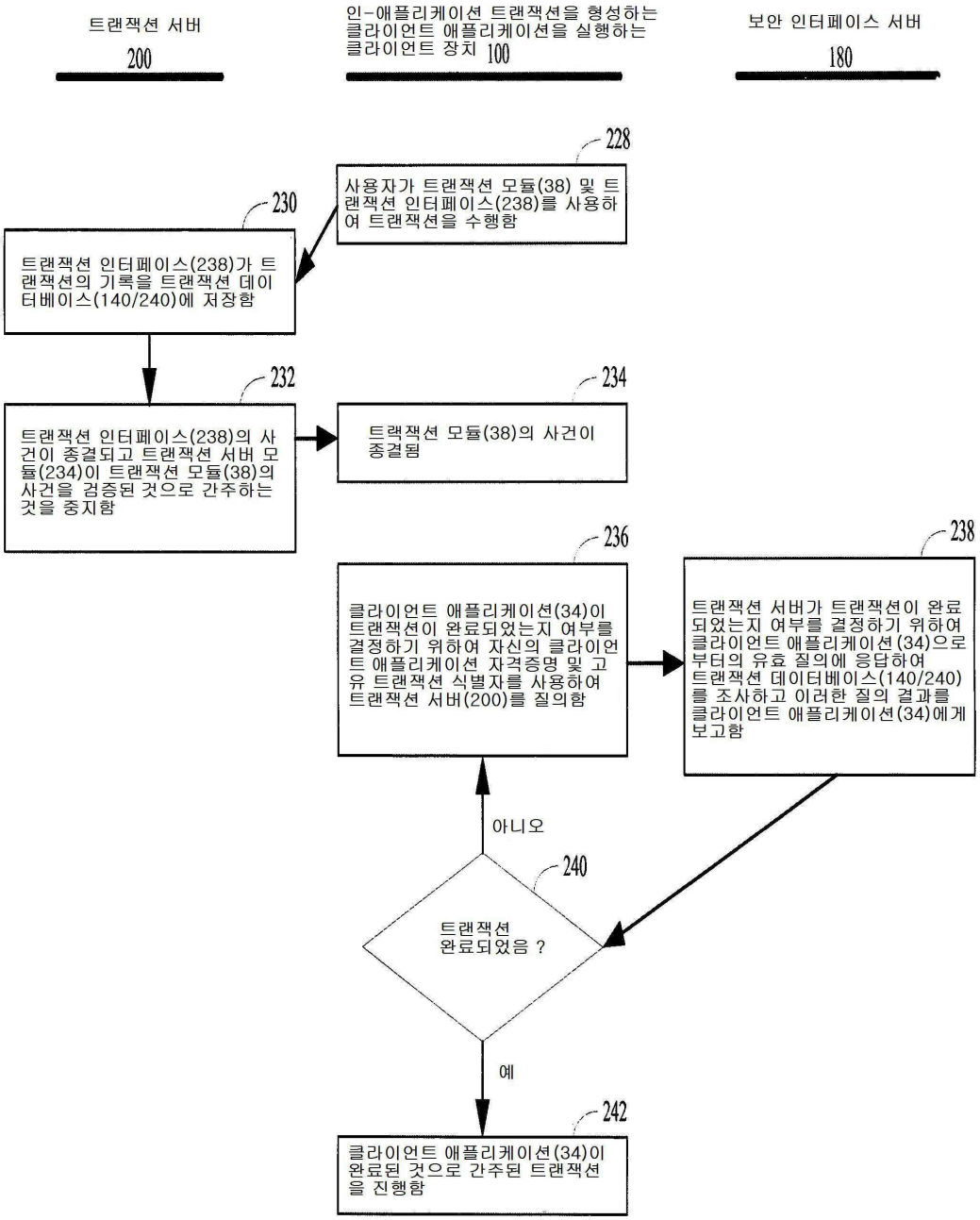
도면1


200
214
운영 체제
230
네트워크 통신 모듈
232
트랜잭션 서버 모듈
234
교차-도메인 정책 2
236
API
238
트랜잭션 데이터베이스
140/240
220
218
제어기
212
210
네트워크 인터페이스
CPU
202
224
전원
공급장치
100
14
운영 체제
30
네트워크 통신 모듈
32
클라이언트 애플리케이션
34
요청 모듈
36
트랜잭션 모듈
38
20
18
제어기
12
4
사용자 인터페이스
디스플레이
6
키보드
8
10
네트워크 인터페이스
CPU
2
24
전원
공급장치
126
인터넷 / 네트워크
180
114
운영 체제
130
네트워크 통신 모듈
132
트랜잭션 모듈 서빙 애플리케이션
134
언브랜디드 트랜잭션 모듈
136
교차-도메인 정책 1
138
트랜잭션 데이터베이스
140/
240
검증된 애플리케이션 자격증명의 데이터베이스
142
120
CPU
102
NIC
110
112
118
제어기
124
전원
공급장치

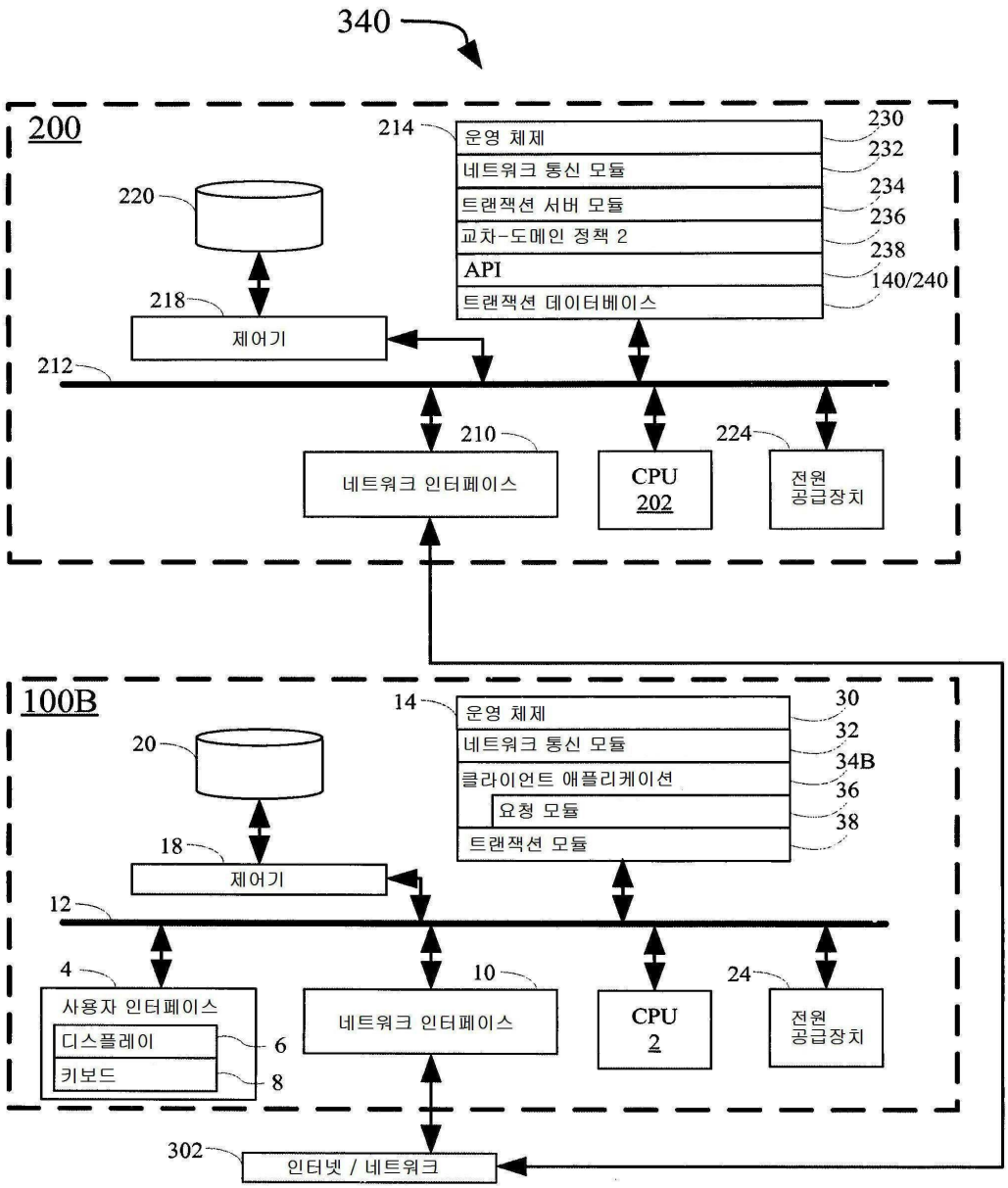
도면2a


트랜잭션 서버
200
인-애플리케이션 트랜잭션을 형성하는 클라이언트 애플리케이션을 실행하는 클라이언트 장치 100
보안 인터페이스 서버
180
202
클라이언트(100)는 로컬 데이터 스토어로부터의 또는 원격 애플리케이션 서버로부터 획득한 클라이언트 애플리케이션(34)을 실행시키고; 클라이언트 애플리케이션(34)은 인-애플리케이션 보안 트랜잭션을 촉진하기 위한 요청 모듈(36)을 포함함
204
클라이언트 애플리케이션(34)은 보안 인-애플리케이션 트랜잭션을 요청하기 위한 요청 모듈(36)을 요구하며; 보안 인-애플리케이션은 애플리케이션 자격증명, 애플리케이션(34) 사용자를 위한 식별하는 정보, 및 고유 트랜잭션 식별자를 포함함
206
보안 인-애플리케이션 트랜잭션에 대한 요청이 애플리케이션 사용자를 식별하는 정보 및 고유 트랜잭션 식별자를 갖는 트랜잭션 모듈 서빙 애플리케이션(134)에 의해 수신됨
208
요청이 제공된 애플리케이션 자격증명이, 검증된 애플리케이션 자격증명의 데이터베이스(142)에 대하여 확인됨
210
확인 ?
212
종료
아니오
예
214
고유 트랜잭션 식별자에 의해 트랜잭션을 트랜잭션 데이터베이스(140/240)에 키(key) 시킴
216
트랜잭션 모듈 서빙 애플리케이션(134)이 요청된 트랜잭션에 대한 트랜잭션 모듈의 검증을 달성하기 위한 충분한 정보로써 언브랜디드 트랜잭션 모듈(예컨대, WWF 파일 포맷)(136)을 다이내믹하게 브랜딩하고 브랜디드 트랜잭션 모듈을 클라이언트 컴퓨터(100)로 트랜잭션 모듈(138)로서 보냄
218
클라이언트 애플리케이션(34)이 보안 인터페이스 서버(180)를 자신들의 소스 URL이라고 식별하는 프로그램에 관련되고 이에 한정되는 별도의 도메인 보안 샌드박스에서 트랜잭션 모듈(38)을 실행하고; 실행된 트랜잭션 모듈(38)은 클라이언트 애플리케이션(34) 또는 요청 모듈(36)에게 내성할 수 있는 허가를 주지 않음
220
트랜잭션 모듈(38)이 트랜잭션 서버(200)에 대한 트랜잭션 호출을 생성함
222
트랜잭션 모듈(38)을 호출하는 것이 보안 인터페이스 서버(180)를 자신의 소스 URL이라고 명명하는 것인가?
아니오
예
224
트랜잭션 실패
226
트랜잭션 서버 모듈(234)이 요청된 트랜잭션을 수행하기 위하여 트랜잭션 인터페이스(238)가 트랜잭션 모듈(38)과 상호대화하는 것을 허용함
228

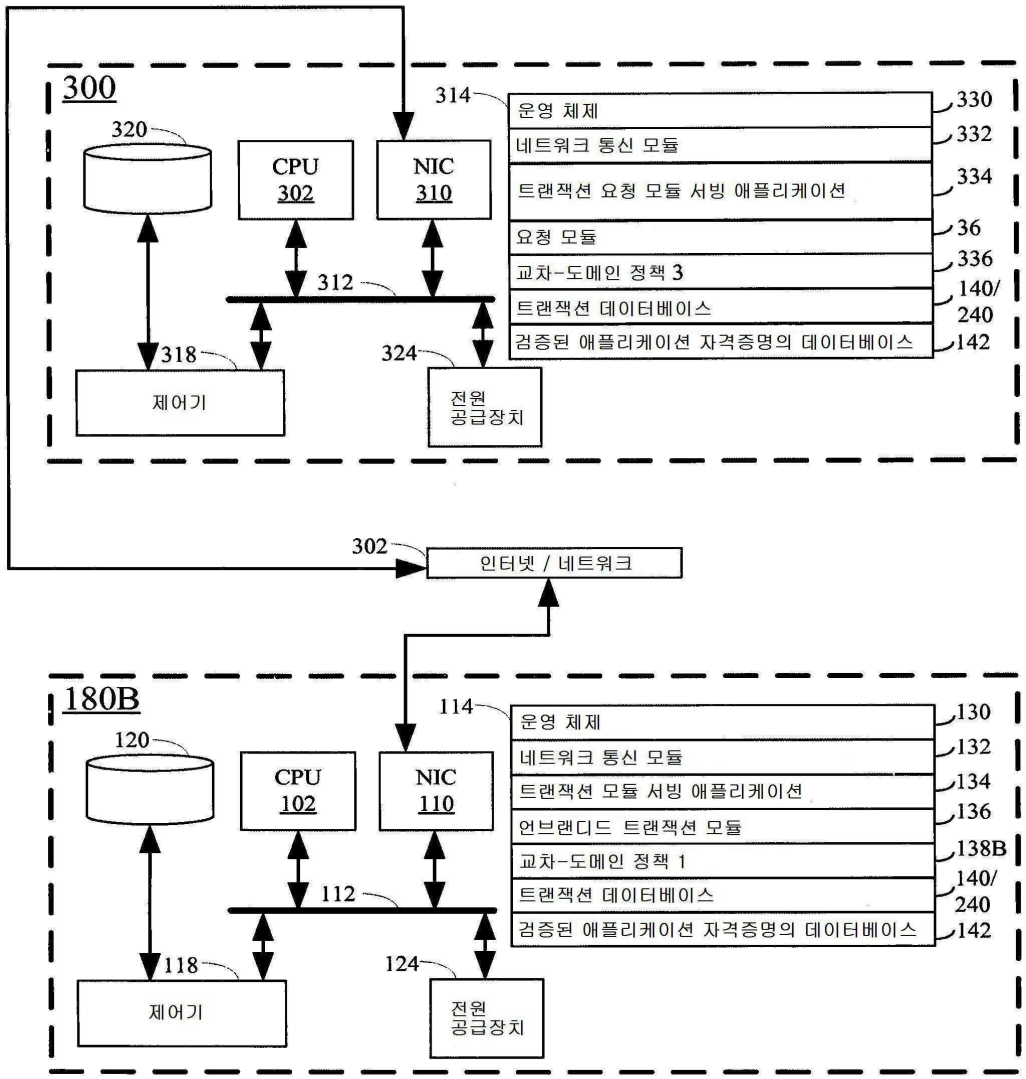
도면2b


트랜잭션 서버
200
인-애플리케이션 트랜잭션을 형성하는 클라이언트 애플리케이션을 실행하는 클라이언트 장치 100
보안 인터페이스 서버
180
228
사용자가 트랜잭션 모듈(38) 및 트랜잭션 인터페이스(238)를 사용하여 트랜잭션을 수행함
230
트랜잭션 인터페이스(238)가 트랜잭션의 기록을 트랜잭션 데이터베이스(140/240)에 저장함
232
트랜잭션 인터페이스(238)의 사건이 종결되고 트랜잭션 서버 모듈(234)이 트랜잭션 모듈(38)의 사건을 검증된 것으로 간주하는 것을 중지함
234
트랙잭션 모듈(38)의 사건이 종결됨
236
클라이언트 애플리케이션(34)이 트랜잭션이 완료되었는지 여부를 결정하기 위하여 자신의 클라이언트 애플리케이션 자격증명 및 고유 트랜잭션 식별자를 사용하여 트랜잭션 서버(200)를 질의함
238
트랜잭션 서버가 트랜잭션이 완료되었는지 여부를 결정하기 위하여 클라이언트 애플리케이션(34)으로부터의 유효 질의에 응답하여 트랜잭션 데이터베이스(140/240)를 조사하고 이러한 질의 결과를 클라이언트 애플리케이션(34)에게 보고함
아니오
240
트랜잭션 완료되었음 ?
예
242
클라이언트 애플리케이션(34)이 완료된 것으로 간주된 트랜잭션을 진행함

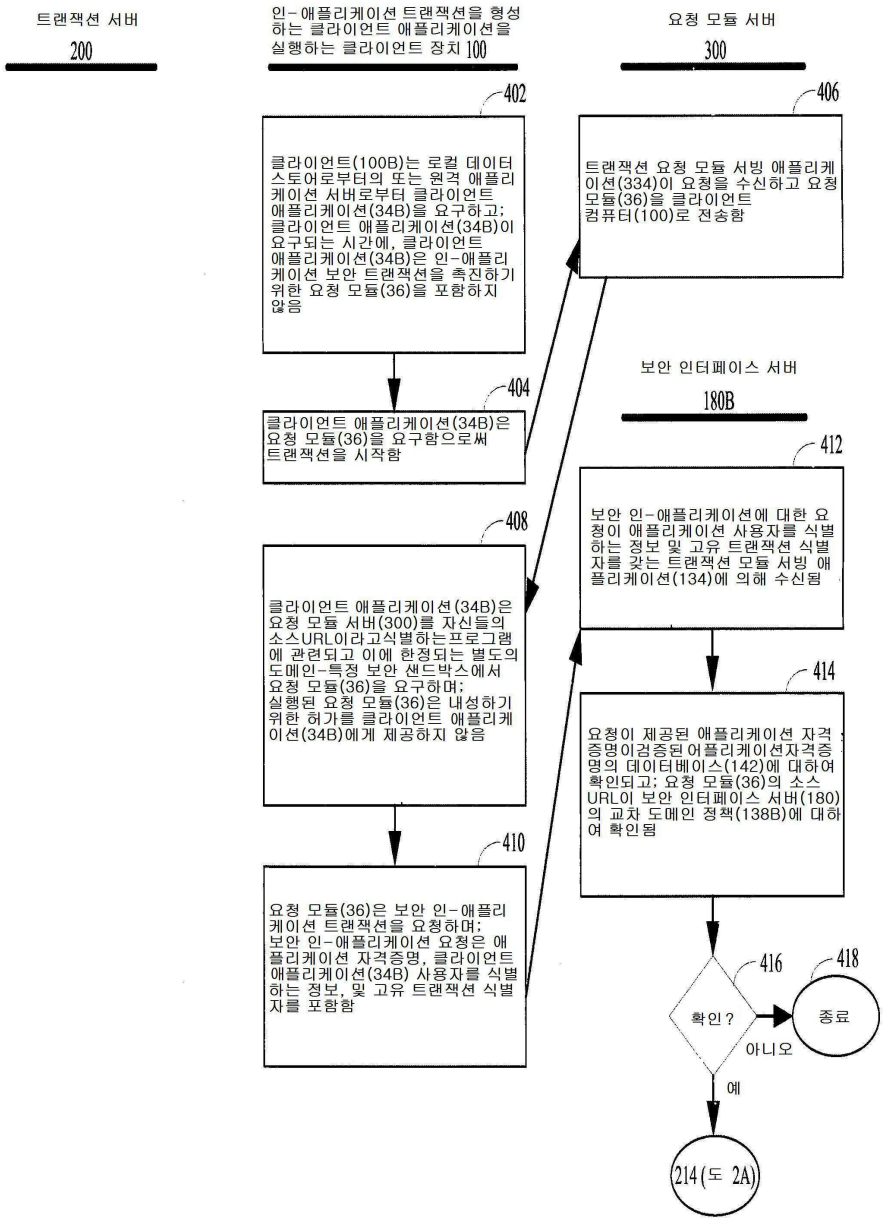
도면3a


340
200
214
운영 체제
네트워크 통신 모듈
트랜잭션 서버 모듈
교차-도메인 정책 2
API
트랜잭션 데이터베이스
230
232
234
236
238
140/240
220
218
제어기
212
210
네트워크 인터페이스
CPU
202
224
전원
공급장치
100B
14
운영 체제
네트워크 통신 모듈
클라이언트 애플리케이션
요청 모듈
트랜잭션 모듈
30
32
34B
36
38
20
18
제어기
12
4
사용자 인터페이스
디스플레이
키보드
6
8
10
네트워크 인터페이스
CPU
2
24
전원
공급장치
302
인터넷 / 네트워크

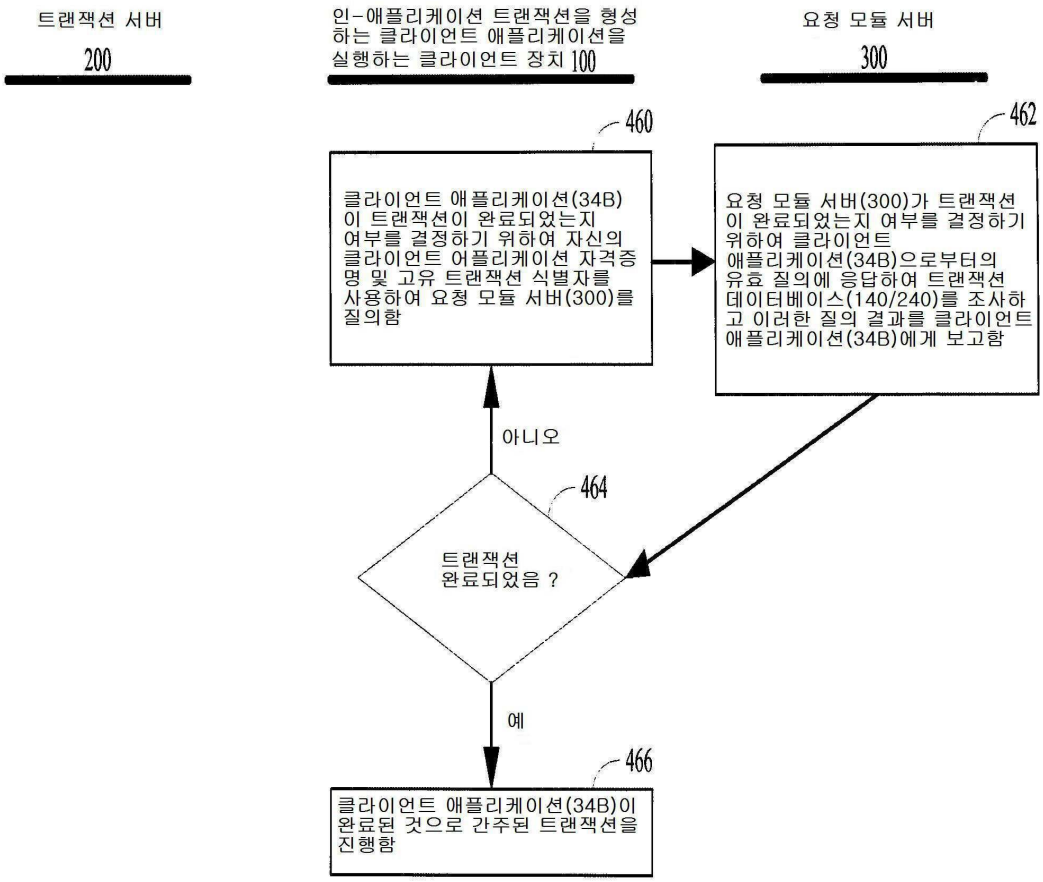
도면3b


300
314
운영 체제
330
네트워크 통신 모듈
332
트랜잭션 요청 모듈 서빙 애플리케이션
334
요청 모듈
36
교차-도메인 정책 3
336
트랜잭션 데이터베이스
140/240
검증된 애플리케이션 자격증명의 데이터베이스
142
320
CPU 302
NIC 310
312
318
제어기
324
전원 공급장치
302
인터넷 / 네트워크
180B
114
운영 체제
130
네트워크 통신 모듈
132
트랜잭션 모듈 서빙 애플리케이션
134
언브랜디드 트랜잭션 모듈
136
교차-도메인 정책 1
138B
트랜잭션 데이터베이스
140/240
검증된 애플리케이션 자격증명의 데이터베이스
142
120
CPU 102
NIC 110
112
118
제어기
124
전원 공급장치

도면4a


트랜잭션 서버
200
인-애플리케이션 트랜잭션을 형성하는 클라이언트 애플리케이션을 실행하는 클라이언트 장치 100
요청 모듈 서버
300
402
클라이언트(100B)는 로컬 데이터 스토어로부터의 또는 원격 애플리케이션 서버로부터 클라이언트 애플리케이션(34B)을 요구하고; 클라이언트 애플리케이션(34B)이 요구되는 시간에, 클라이언트 애플리케이션(34B)은 인-애플리케이션 보안 트랜잭션을 촉진하기 위한 요청 모듈(36)을 포함하지 않음
406
트랜잭션 요청 모듈 서빙 애플리케이션(334)이 요청을 수신하고 요청 모듈(36)을 클라이언트 컴퓨터(100)로 전송함
404
클라이언트 애플리케이션(34B)은 요청 모듈(36)을 요구함으로써 트랜잭션을 시작함
보안 인터페이스 서버
180B
412
보안 인-애플리케이션에 대한 요청이 애플리케이션 사용자를 식별하는 정보 및 고유 트랜잭션 식별자를 갖는 트랜잭션 모듈 서빙 애플리케이션(134)에 의해 수신됨
408
클라이언트 애플리케이션(34B)은 요청 모듈 서버(300)를 자신들의 소스URL이라고식별하는프로그램에 관련되고 이에 한정되는 별도의 도메인-특정 보안 샌드박스에서 요청 모듈(36)을 요구하며; 실행된 요청 모듈(36)은 내성하기 위한 허가를 클라이언트 애플리케이션(34B)에게 제공하지 않음
414
요청이 제공된 애플리케이션 자격증명이검증된어플리케이션자격증명의 데이터베이스(142)에 대하여 확인되고; 요청 모듈(36)의 소스 URL이 보안 인터페이스 서버(180)의 교차 도메인 정책(138B)에 대하여 확인됨
410
요청 모듈(36)은 보안 인-애플리케이션 트랜잭션을 요청하며; 보안 인-애플리케이션 요청은 애플리케이션 자격증명, 클라이언트 애플리케이션(34B) 사용자를 식별하는 정보, 및 고유 트랜잭션 식별자를 포함함
416
확인?
418
종료
아니오
예
214(도 2A)

도면4b


트랜잭션 서버
200
인-애플리케이션 트랜잭션을 형성하는 클라이언트 애플리케이션을 실행하는 클라이언트 장치 100
요청 모듈 서버
300
460
클라이언트 애플리케이션(34B)이 트랜잭션이 완료되었는지 여부를 결정하기 위하여 자신의 클라이언트 어플리케이션 자격증명 및 고유 트랜잭션 식별자를 사용하여 요청 모듈 서버(300)를 질의함
462
요청 모듈 서버(300)가 트랜잭션이 완료되었는지 여부를 결정하기 위하여 클라이언트 애플리케이션(34B)으로부터의 유효 질의에 응답하여 트랜잭션 데이터베이스(140/240)를 조사하고 이러한 질의 결과를 클라이언트 애플리케이션(34B)에게 보고함
아니오
464
트랜잭션 완료되었음 ?
예
466
클라이언트 애플리케이션(34B)이 완료된 것으로 간주된 트랜잭션을 진행함