



(12) **PATENT**

(19) **NO**

(11) **331932**

(13) **B1**

NORGE

(51) **Int Cl.**

G06F 12/00 (2006.01)

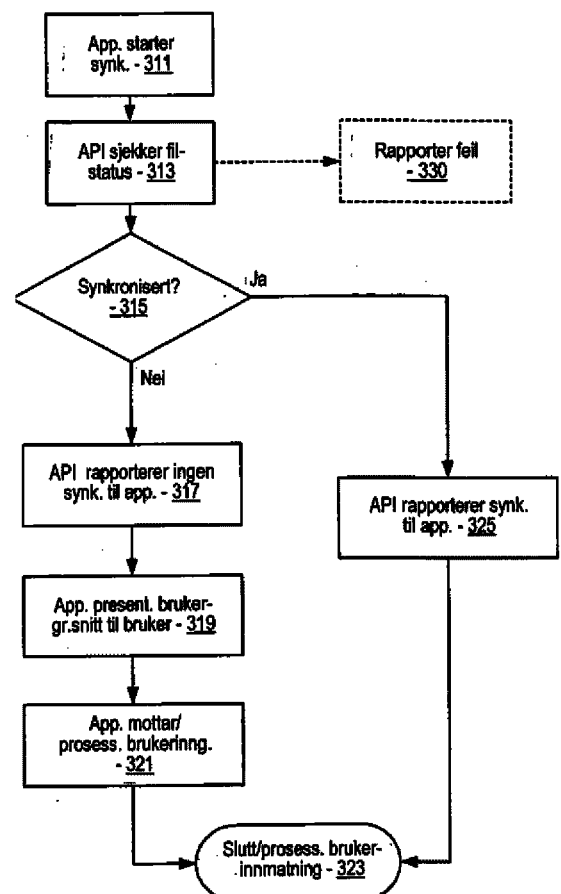
G06F 17/22 (2006.01)

G06F 9/44 (2006.01)

Patentstyret

(21)	Søknadsnr	20040834	(86)	Int.inng.dag og søknadsnr	
(22)	Inng.dag	2004.02.25	(85)	Videreføringsdag	
(24)	Løpedag	2004.02.25	(30)	Prioritet	2003.02.28, US, 376738
(41)	Alm.tilgj	2004.09.01			
(45)	Meddelt	2012.05.07			
(73)	Innehaver	Microsoft Corp, One Microsoft Way, US-WA98052-6399 REDMOND, USA			
(72)	Oppfinner	Noah Edelstein, 360 North 26th Street, US-WA98103 SEATTLE, USA Hani Saliba, 524 12th Avenue East, #302, US-WA98102 SEATTLE, USA Hai Liu, 4850 156th Avenue, Apartment #280, US-WA98052 REDMOND, USA Steven James Rayson, 609 Federal Avenue East, US-WA98102 SEATTLE, USA			
(74)	Fullmektig	Bryn Aarflot AS, Postboks 449 Sentrum, 0104 OSLO, Norge			
(54)	Benevnelse	Fremgangsmåte for styring av fil-replikering i applikasjoner.			
(56)	Anførte publikasjoner	US 6,449,624 B1			
(57)	Sammendrag				

En fremgangsmåte for å styre en første og andre versjon av en elektronisk fil er gjort kjent. I fremgangsmåten blir første og andre versjon av en elektronisk fil gjenkjent og en synkroniserings-hendelse blir startet. Et applikasjons-programgrensesnitt har tilgang til den første og andre versjon, sammenlikner informasjon i samsvar med hver av dem, frembringer et resultat, og rapporterer resultatet til den første programvareapplikasjon. En andre programvareapplikasjon frembringer et brukergrensesnitt i henhold til et trekk i den andre programvareapplikasjon, hvor den elektroniske fil er assosiert med den andre programvareapplikasjon, og brukergrensesnittet blir vist på en visningsanordning.



OPPFINNELSENS OMRÅDE

Den foreliggende oppfinnelse er rettet mot fremgangsmåter for styring av flere versjoner av en elektronisk fil. Mer spesielt, angår den foreliggende oppfinnelse fremgangsmåter for bruk av relevant applikasjons-funksjonalitet for å forene
5 filversjoner under en synkroniseringshendelse.

BAKGRUNN FOR OPPFINNELSEN

Det er vanlig praksis i et nettverksforbundet datamaskinmiljø at en datamaskinbruker samarbeider på et elektronisk dokument med en annen bruker på
10 nettverket. Et slikt nettverk kan være, et lokalt nettverk (LAN), et bredbåndet nettverk (WAN), internett eller liknende. I et slikt samarbeid kan hver bruker opprette eller redigere en elektronisk fil ved bruk av en programvareapplikasjon slik som generell applikasjons-programvare, eller programvare spesialutviklet for filtypen som det samarbeides på, eller liknende.

15 Samarbeidet kan gjøres mulig ved å vedlikeholde en hovedversjon av den elektroniske fil på en lokasjon som er tilgjengelig for alle brukere som samarbeider på filen. I et konvensjonelt nettverksforbundet datamaskinmiljø, kan det være en tjener-datamaskin og en eller flere bruker-datamaskiner. Tjeneren opprettholder kommunikasjonsforbindelser mellom brukerne, og er tilgjengelig for hver bruker-
20 datamaskin, og kan derfor lagre og kontrollere hovedversjonen av filen som brukerne samarbeider på. Ved samarbeid på en elektronisk fil, kan en bruker først aksessere hovedversjonen av filen på tjeneren, og deretter kopiere filen til brukerens datamaskin. Brukeren kan så arbeide på filen ved å endre på filen på en hvilken som helst måte gjort tilgjengelig for brukeren, av applikasjonen. For eksempel,
25 hvis applikasjonen er et tekstbehandlingsprogram, regneark, database, og/eller liknende, kan applikasjonen tilveiebringe hjelpemidler for utføring av oppgaver slik som redigering, kopiering, sletting og/eller liknende. Ved fullføring av enhver endring, kan brukeren lagre endringene til fil-versjonen lagret på brukerens datamaskin hvis brukeren ønsker å beholde disse endringene på filen. På andre systemer, slik som et system hvor ingen tjener-datamaskin finnes, kan hovedversjonen
30 lagres på en bruker-datamaskin.

For å kunne samarbeide med andre brukere via tjeneren, må en bruker med en endret versjon lagret på brukerens datamaskin, synkronisere filversjonen på datamaskinen med hovedversjonen av filen på tjeneren. Synkronisering av et

flertall av versjoner av en elektronisk fil, i sin enkleste form, forsikrer at alle versjoner, etter synkronisering, er like. På denne måten kan brukeren som utfører synkroniseringen muliggjøre at andre brukere som deltar i samarbeidet kan ha tilgang til den ny-oppdaterte filen. Konvensjonelt er en synkronisering utført av et synkroniseringsverktøy. Et synkroniseringsverktøy er vanligvis separert fra applikasjonen, og er på mange områder en del av operativsystemet på brukerens datamaskin. Det konvensjonelle synkroniseringsverktøyet utfører synkronisering på forhåndsbestemte tider, eller på forespørsel fra operativsystemet.

I en konvensjonell synkronisering sammenlikner et synkroniseringsverktøy brukerversjonen av filen med hovedversjonen av filen ved bruk av enhver informasjon om kopiene som verktøyet er i stand til å samle. Vanligvis er slik informasjon begrenset til datoen og/eller tiden på den siste oppdatering av hver versjon. Hvis datoen og/eller tiden er lik, bestemmer synkroniseringsverktøyet at begge versjoner er like, og derfor synkronisert, og at ingen videre handling er nødvendig. Hvis det derimot eksisterer en uoverensstemmelse mellom datoen og/eller tiden, tilveiebringer synkroniseringsverktøyet brukeren, som er assosiert med datamaskinen som styrer synkroniseringshendelsen, med et brukergrensesnitt. Brukergrensesnittet tilveiebringer vanligvis en bruker med en kort beskjed om at det er en uoverensstemmelse, datoen og/eller tiden for siste oppdatering av hver filversjon, og en forespørsel om en instruksjon fra brukeren om hvilken kopi av filen som brukeren ønsker å beholde. Hvis brukeren velger å beholde brukerversjonen, overskriver synkroniseringsverktøyet hovedversjonen med brukerversjonen, for dermed å gjøre brukerversjonen til den nye hovedversjonen på tjeneren. Hvis brukeren ønsker å beholde hovedversjonen, tas ingen flere valg, eller brukerversjonen overskrives med hovedversjonen. I noen brukergrensesnitt blir brukeren presentert med et valg for å avbryte synkroniseringen, istedenfor å velge hvilken versjon som skal være hovedversjonen.

Den konvensjonelle synkroniseringen som diskutert ovenfor har betydelige ulemper. Siden et konvensjonelt synkroniseringsverktøy har begrenset funksjonalitet i forhold til en elektronisk fil, presenterer et brukergrensesnitt generert av det konvensjonelle verktøy bare en begrenset mengde informasjon til brukeren når man forespør en bruker om å velge en handlingsvei. En slik begrenset mengde informasjon kan derfor forårsake at en slik bruker tar en utilstrekkelig og muligens feil avgjørelse om hvilken handlingsvei som skal tas. For eksempel kan en første

bruker endre en tidligere synkronisert brukerversjon av filen, og deretter på et tidspunkt starter det konvensjonelle synkroniserings-verktøyet en synkroniseringshendelse. Ved underrettelse av et konvensjonelt brukergrensesnitt, kan brukeren se at brukerversjonen ble oppdatert senere enn hovedversjonen og derfor bestemme seg for å overskrive hovedversjonen. Informasjonen vist av brukergrensesnittet informerer derimot ikke brukeren om hovedversjonen er forblitt uforandret siden brukeren begynte å endre brukerversjonen. Dette fører til en problematisk situasjon, hvor en andre bruker kan ha gjort endringer på en annen brukerversjon, og så synkronisert denne brukerversjonen med hovedversjonen før den første brukers synkronisering. Den første brukers versjon er nyere - og er blitt oppdatert senere - men hovedversjonen har undergått endringer av den andre bruker som ikke var tilstede da den første bruker først kopierte den originale filen fra tjeneren. Dette resulterer i at den andre brukers endringer på hovedversjonen går tapt når den første bruker fullfører synkroniseringen.

Situasjonen ovenfor er forårsaket av et konvensjonelt synkroniseringsverktøys mangler når det gjelder hele funksjonalitetsområdet iboende applikasjonen relevant til filen, så verktøyet er ikke i stand til å bruke denne funksjonaliteten under en synkroniseringshendelse. Verktøyet kan derfor bare tilveiebringe en bruker med en meget begrenset mengde informasjon angående de forskjellige versjoner av den elektroniske filen, slik som f.eks. datoen og/eller tiden for den siste oppdatering av hver filversjon som diskutert ovenfor.

En annen ulempe forårsaket av den begrensede funksjonaliteten i konvensjonelle synkroniseringsverktøy, er mangelen på valgmuligheter tilgjengelig for en bruker under en synkronisering. For eksempel, i eksempelet diskutert ovenfor, presenterer brukergrensesnittet bare tre valg: oppdater hovedkopien med brukerkopien, oppdater brukerkopien med hovedkopien, eller avbryt.

En videre ulempe med et konvensjonelt synkroniseringsverktøy skyldes mangelen på integrasjon av slike verktøy med slike applikasjoner. For eksempel, og som diskutert ovenfor, er konvensjonelle synkroniseringsverktøy en del av et operativsystem på en bruker- eller tjener-datamaskin. Synkroniseringsprosessen og ethvert anvendelig brukergrensesnitt er derfor vanligvis ledet utenfor applikasjon-bruker erfaringene. Dette kan føre til forvirring blant brukerne, som, selv om slike brukere kan være dyktige på applikasjonen, så er de ikke dyktige med et operativsystem-verktøy. I tillegg, og som et resultat av mangelen på inte-

grasjon, kan en synkronisering startes av synkroniseringsverktøyet på et tidspunkt som ikke er optimalt for en hendelse av denne typen. En ytterligere ulempe med konvensjonelle synkroniseringsverktøy er den manglende evnen slike verktøy har til å kunne brukes som en komponent i en programvareapplikasjon. For eksempel, ved bruk av konvensjonelle metoder, er ikke en dataprogrammerer i stand til å utvikle en første programvareapplikasjon som bruker et synkroniseringsverktøy som har evnen til å bruke en andre applikasjon som er forbeholdt den elektroniske fil, for å forene flere versjoner av denne elektroniske filen. Det som behøves er derfor en effektiv måte å bruke relevant programvareapplikasjon-funksjonalitet på, for å løse konflikter mellom filkopier under en synkronisering.

US 6449624 angår generelt prosesskontrollsystemer, og mer spesielt et system som overvåker og registrerer modifiseringer til prosessen for å tilveiebringe versjonskontroll. Systemet inkluderer et datamaskinlesbart medium, en prosessor i kommunikasjon med det datamaskinlesbare medium, og en første og en andre database. Den første databasen lagrer første data representative for en første konfigurering av prosessen, mens den andre databasen lagrer andre data representative for en andre konfigurering av prosessen. En konfigurasjonsrutine av systemet er lagret i det datamaskinlesbare medium og konfigurert til å utføres av prosessoren for å lette en modifisering av den første konfigurasjonsprosessen. En versjonskontrollrutine for systemet er lagret i det datamaskinlesbare medium og vurderes å bli utført av prosessoren for å lagre i den andre databasen tredje data indikative på modifiseringen av den første konfigurasjonen av prosessen.

SAMMENFATNING AV OPPFINNELSEN

I et første aspekt tilveiebringer den foreliggende oppfinnelsen en fremgangsmåte for synkronisering av en første og en andre versjon av en elektronisk fil, hvor fremgangsmåten omfatter: å starte, ved en første programvareapplikasjon som opererer på en første datamaskin, en synkroniserings-hendelse for å synkronisere den første versjonen med den andre versjonen; å forårsake, ved den første programvareapplikasjonen at et applikasjons-programgrensesnitt aksesserer nevnte første og andre versjoner; å sammenlikne, ved applikasjonsprogramgrensesnittet, informasjon assosiert med hver av de nevnte første og andre versjoner for å bestemme hvorvidt den første og den andre versjonen er synkronisert, og

hvis ikke: å rapportere, ved applikasjonsprogramgrensesnittet, til den første programvareapplikasjonen at den første og den andre versjonen ikke er synkronisert; å forårsake at en andre programvareapplikasjon frembringer et brukergrensesnitt i henhold til et trekk i den andre programvareapplikasjonen, hvor den elektroniske fil
5 er i samsvar med den andre programvareapplikasjon; og å vise brukergrensesnittet som presenterer én eller flere synkroniseringsmuligheter for å synkronisere den første og den andre versjonen på en visningsanordning; å motta, ved den andre programvareapplikasjonen, et valg av én av de én eller flere synkroniseringsmuligheter fra en bruker via brukergrensesnittet; og å synkronisere første og andre versjoner i henhold til den valgte synkroniseringsmulighet.
10

Foretrukkede utførelsesformer av oppfinnelsen er definert i de uselvstendige krav 2-13.

I et andre aspekt tilveiebringer oppfinnelsen et datamaskin-lesbart medium som har datamaskin-eksekverbare instruksjoner lagret derpå som når de utføres
15 av en eller flere datamaskiner forårsaker den ene eller flere datamaskinen å utføre en fremgangsmåte i henhold til oppfinnelsen angitt over.

Det beskrives en fremgangsmåte for å integrere en applikasjons programgrensesnitt med en programvareapplikasjon som muliggjør at applikasjonen kan forespørre og kontrollere en filsynkroniseringsprosess, innenfor applikasjonens
20 grenser og brukers erfaringer. Oppfinnelsen tilveiebringer en fremgangsmåte for styring av en første og andre versjon av en elektronisk fil. I fremgangsmåten blir første og andre versjon av en elektronisk fil gjenkjent, og en synkroniseringsprosess blir startet ved bruk av en første programvareapplikasjon som opererer på en første datamaskin. Et applikasjons programgrensesnitt aksesserer den første og
25 andre versjonen, sammenlikner informasjon assosiert med hver av dem, frembringer et resultat og rapporterer resultatet til den første programvareapplikasjonen. En andre programvareapplikasjon frembringer et brukergrensesnitt i henhold til et trekk i den andre programvareapplikasjonen, hvor den elektroniske fil blir assosiert med den andre programvareapplikasjonen, og brukergrensesnittet blir vist
30 på en visningsanordning.

Det beskrives også en fremgangsmåte for den andre programvareapplikasjonen for å frembringe et synkroniseringsvalg og å presentere valget sammen med brukergrensesnittet og å vise brukergrensesnittet på visningsanordningen. Den andre programvareapplikasjonen kan også motta en valgselektering og syn-

kroniserer den første og andre versjon av den elektroniske fil i henhold til dette. Det beskrives videre at den første programvareapplikasjonen og den andre programvareapplikasjonen kan være den samme programvareapplikasjonen. Applikasjonen kan tilveiebringe enhver informasjonsmengde til brukeren når det gjelder
 5 den første og andre versjonen, og brukeren kan taste inn et hvilket som helst antall eller type synkroniseringsvalg. I en utførelsesform kan brukeren opprette en tredje versjon som omfatter en hvilken som helst del, eller ingen, av enten første eller andre versjon, eller begge. Applikasjonen kan starte synkroniseringen når som helst, slik som ved forespørsel fra brukeren. Ytterligere trekk og fordeler med
 10 oppfinnelsen vil fremgå av beskrivelsen nedenfor.

KORT BESKRIVELSE AV TEGNINGENE

Den forannevnte sammenfatning, i tillegg til den følgende detaljerte beskrivelse av foretrukne utførelsesformer, kan bedre forstås når den leses sammen
 15 med de vedføyde tegninger. For formålet med å illustrere oppfinnelsen, er det på tegningene vist eksempler på utførelsesformer av oppfinnelsen; derimot er ikke oppfinnelsen begrenset til de spesielle fremgangsmåter og hjelpemidler som er avdekket.

Fig. 1 er et blokkdiagram som viser et eksempel på et datamaskinmiljø hvor aspektene i oppfinnelsen kan implementeres.
 20

fig. 2 er et funksjonsdiagram som illustrerer et eksempel på et forhold mellom en første datamaskin, et nettverk og en andre datamaskin.

fig. 3 er et flytdiagram som viser en synkroniseringsprosess i henhold til en utførelsesform av den foreliggende oppfinnelse; og

25 fig. 4 er et flytdiagram som illustrerer en fremgangsmåte for å starte en synkroniseringsprosess i henhold til en utførelsesform av den foreliggende oppfinnelse.

DETALJERT BESKRIVELSE AV ILLUSTRERENDE UTFØRELSESFORMER

30 Oversikt

Den foreliggende oppfinnelse tilveiebringer fremgangsmåter for å integrere et applikasjon-programgrensesnitt med en programvareapplikasjon, som gjør det mulig for applikasjonen å forespørre og kontrollere en filsynkroniseringsprosess innenfor applikasjonens brukererfaringer. Oppfinnelsen tilveiebringer en frem-

gangsmåte for styring av en første og andre versjon av en elektronisk fil. I fremgangsmåten blir første og andre versjon av en elektronisk fil gjenkjent og en synkroniseringshendelse for å synkronisere den første versjonen på den første datamaskin med den andre versjon, blir startet ved bruk av en første programvareapplikasjon som opererer på en første datamaskin. Et applikasjon-programgrensesnitt
 5 aksesserer den første og andre versjonen, informasjon assosiert med hver av den første og andre versjonen blir sammenliknet, og et resultat blir tilveiebrakt. Resultatet av sammenlikningen rapporteres til den første programvareapplikasjonen. En andre programvareapplikasjon forårsaker frembringelse av et brukergrensesnitt i
 10 henhold til et trekk i den andre programvareapplikasjonen, hvor den elektroniske fil blir assosiert med den andre programvareapplikasjonen, og brukergrensesnittet blir vist på en visningsanordning.

Oppfinnelsen sørger også for at den andre programvareapplikasjonen frembringer et synkroniseringsvalg og forårsaker at synkroniseringsvalget blir presentert av brukergrensesnittet og vist på visningsanordningen. Den andre programvareapplikasjon mottar også en valgselektering og synkroniserer den første og andre versjon av den elektroniske fil i henhold til valgselekteringen. Oppfinnelsen sørger videre for at den første programvareapplikasjon og den andre programvareapplikasjon kan være den samme programvareapplikasjonen, og alle trinn kan
 20 derfor bli utført av en enkel applikasjon.

Applikasjonen kan fremskaffe en hvilken som helst informasjonsmengde til brukeren angående den første og andre versjonen, og brukeren kan taste inn et hvilket som helst antall eller type synkroniseringsvalg. I en utførelsesform kan brukeren opprette en tredje versjon som omfatter en hvilken som helst del, eller ingen, av enten første eller andre versjon, eller begge. Applikasjonen kan starte synkroniseringen på et hvilket som helst tidspunkt, slik som på brukerens forespørsel.
 25

EKSEMPLER PÅ DATAMASKIN MILJØ

Fig. 1 illustrerer et eksempel på et passende datamaskinmiljø 100 hvor oppfinnelsen kan implementeres. Datamaskinmiljøet 100 er bare ett eksempel på et
 30 passende datamaskinmiljø og er ikke ment for å foreslå noen begrensninger på oppfinnelsens bruksomfang eller funksjonalitet. Datamaskinmiljøet skal heller ikke tolkes som om det er avhengig av eller har krav relatert til hvilke som helst eller

kombinasjon av komponenter som illustrert i det eksemplifiserte operasjonsmiljøet 100.

Oppfinnelsen er operativ med flere andre generelle formål eller spesialformål, datasystemmiljøer eller konfigurasjoner. Eksempler på velkjente
5 datamaskin-systemer, miljøer, og/eller konfigurasjoner som kan være passende for bruk sammen med oppfinnelsen innbefatter, men er ikke begrenset til, personlige datamaskiner, tjener-datamaskiner, håndholdte eller bærbare anordninger, flerprosessor-systemer, mikroprosessor-baserte systemer, "set top boxes", programmerbar bruker-elektronikk, nettverks-PC'er, mikrodatamaskiner, stormaski-
10 ner, distribuerte datamiljøer som inkluderer en hvilken som helst av systemene eller anordningene ovenfor, eller liknende.

Oppfinnelsen kan beskrives i generell sammenheng med datamaskin-eksekverbare instruksjoner, slik som programmoduler som blir eksekvert av en datamaskin. Generelt inneholder programmodulene rutiner, programmer, objekter,
15 komponenter, datastrukturer, osv. som utfører spesielle oppgaver eller implementerer bestemte abstrakte datatyper. Oppfinnelsen kan også anvendes i distribuerte datamiljøer hvor oppgaver blir utført av fjerntliggende prosesserende anordninger som er forbundet gjennom et kommunikasjonsnettverk eller annet dataoverføringsmedium. I et distribuert datamiljø kan programmoduler og annen data være lokali-
20 sert på både lokale og fjerntliggende datalagringsmedier, innbefattende minnelagringsanordninger.

Med referanse til fig. 1, innbefatter et system-eksempel for implementering av oppfinnelsen en dataanordning for generelle formål, i form av en datamaskin 110. Komponentene i datamaskin 110 kan innbefatte, men er ikke begrenset til,
25 en prosesseringsenhet 120, et systemminne 130, og en systembuss 121 som kobler forskjellige systemkomponenter, inkludert systemminne til prosesseringsenheten 120. Systembussen 121 kan være en hvilken som helst av flere typer buss-strukturer innbefattende en minnebuss eller minnekontroller, en ytre buss, og en lokalbuss som bruker en hvilken som helst av en varietet av bussarkitekturer. Som
30 et eksempel, og ikke med begrensning, innbefatter slike arkitekturer, Industri Standard Architecture (ISA) -buss, Micro Channel Architecture (MCA) -buss, Enhanced ISA (EISA) -buss, Video Electronic Standards Association (VESA) -lokalbuss, og Peripheral Component InterConnect (PCI) -buss (også kjent som Mezzanine-buss).

Datamaskin 110 innbefatter vanligvis en varietet av datamaskin-lesbare medier. Datamaskin-lesbare medier kan være ethvert tilgjengelig medium som kan bli aksessert av datamaskin 110 og inkluderer både flyktige og ikke-flyktige medier, flyttbare og ikke-flyttbare medier. Som et eksempel, og ikke med begrensning, kan datamaskin-lesbare medier omfatte datamaskin-lagringsmedier og kommunikasjonsmedier. Datamaskin-lagringsmedier innbefatter både flyktige og ikke-flyktige, flyttbare og ikke-flyttbare medier implementert på en hvilken som helst metode eller teknologi for lagring av informasjon, slik som datamaskin-lesbare instruksjoner, datastrukturer, programmoduler eller annen data.

Datamaskin-lagringsmedier inkluderer, men er ikke begrenset til, RAM, ROM, EEPROM, flash-minne eller annen minne-teknologi, CD-ROM, "digital versatile disks" (DVD) eller annet optisk disklager, magnetiske kassetter, magnetiske bånd, magnetisk disklager eller annen magnetisk lagringsanordning, eller ethvert annet medium som kan brukes til å lagre den ønskede informasjon og som kan aksesserres av datamaskin 110. Kommunikasjonsmedier innlemmer vanligvis datamaskin-lesbare instruksjoner, datastrukturer, programmoduler eller annen data i et modulert datasignal, slik som en bæreølge eller annen frakte-mekanisme, og innbefatter ethvert informasjonsoverleverings-medium. Med uttrykket "modulert datasignal" menes et signal som får en eller flere av sine karakteristikk satt eller forandret på en måte slik at den koder informasjon i signalet. Som et eksempel, og ikke med begrensning, innbefatter kommunikasjonsmedier, koblede medier slik som et koblet nettverk eller direkte-koblet forbindelse, og trådløse medier slik som akustikk, RF, infrarødt og annet trådløst medium. Kombinasjoner av en hvilken som helst av de ovenstående skal også innbefattes innen omfanget av datamaskin-lesbare medier.

System-minnet 130 innbefatter datamaskin-lagringsmedier i form av flyktig og/eller ikke-flyktig minne, slik som ROM 131 og RAM 132. Et standard inngangs-/utgangs-system 133 (BIOS), som inneholder standard-rutiner som hjelper til med å overføre informasjon mellom elementer innenfor datamaskin 110, slik som ved oppstartning, er vanligvis lagret i ROM 131. RAM 132 inneholder typisk data og/eller programmoduler som for øyeblikket er tilgjengelig for, og/eller for øyeblikket blir arbeidet på, av prosesseringsenhet 120. Som et eksempel, og ikke med begrensning, illustrerer fig. 1 operativsystem 134, applikasjonsprogrammer 135, andre programmoduler 136, og programdata 137.

Datamaskin 110 kan også innbefatte andre flyttbare/ikke-flyttbare, flyktige/ikke-flyktige datamaskin-lagringsmedier. Bare som et eksempel, illustrerer fig. 1 en harddisk 140 som leser fra eller skriver til ikke-flyttbart, ikke-flyktig magnetisk medium, en magnetisk disk stasjon 151 som leser fra eller skriver til en flyttbar, ikke-flyktig magnetisk disk 152, og en optisk disk stasjon 155 som leser fra eller skriver til en flyttbar, ikke-flyktig optisk disk 156, slik som en CD-ROM eller annet optisk medium. Andre flyttbare/ikke-flyttbare, flyktige/ikke-flyktige datamaskin-lagringsmedier som kan bli brukt i det eksempelvis operasjonsmiljøet innbefatter, men er ikke begrenset til, magnetiske bånd-kassetter, flashminne-kort, digital versatile disks (DVD), digitale video- bånd, fastminne, fast-ROM, og liknende. Harddisken 141 er typisk koblet til systembussen 121 gjennom et ikke-flyttbart minnegrensesnitt, slik som grensesnitt 140, og magnetiskdisk stasjon 151 og optisk diskstasjon 155 er typisk koblet til systembussen 121 av et flyttbart minnegrensesnitt, slik som grensesnitt 150.

Stasjonene og deres tilhørende datamaskin-lagringsmedier, diskutert ovenfor og illustrert i fig. 1, tilveiebringer lagring av datamaskin-lesbare instruksjoner, datastrukturer, programmoduler og annen data for datamaskinen 110. I fig. 1, for eksempel, er harddisk 141 illustrert som lagrer operativ-system 144, applikasjonsprogrammer 145, andre programmoduler 146, og programdata 147. Bemerk at disse komponentene enten kan være de samme som eller forskjellig fra operativ-system 134, applikasjonsprogrammer 135, andre programmoduler 136, og programdata 137. Operativsystem 144, applikasjonsprogrammer 145, andre programmoduler 146, og programdata 147 er her gitt forskjellige numre for å illustrere at, i det minste, de er forskjellige kopier. En bruker kan skrive inn kommandoer og informasjon til datamaskinen 110 gjennom inngangsanordninger slik som et tastatur 162 og pekeanordning 161, vanligvis referert til som en mus, "track-ball" eller berøringspute (touch pad). Andre inngangsanordninger (ikke vist) kan innbefatte en mikrofon, joystick, spillkontroller, satellitt-tallerken, skanner, eller liknende. Disse og andre inngangsanordninger er ofte koblet til prosesseringsenheten 120 gjennom et brukergrensesnitt 160 som er koblet til systembussen, men kan være koblet ved annet grensesnitt og buss-strukturer, slik som en parallellport, game-port eller en universell seriell buss (USB). En monitor 19 eller annen type visningsanordning er også koblet til systembussen 121 via et grensesnitt, slik som et videogrensesnitt 190. I tillegg til monitoren, kan datamaskiner også innbefatte andre

periferiske utgangsanordninger, slik som høyttalere 197 og skriver 196, som kan kobles gjennom et utgangsgrensesnitt 190.

Datamaskinen 110 kan operere i et nettverksmiljø som bruker logiske forbindelser til en eller flere fjerntliggende datamaskiner, slik som en fjerntliggende datamaskin 180. Den fjerntliggende datamaskin 180 kan være en personlig data-
 5 maskin, en tjener, en ruter, en nettverks-PC, en endepunktsanordning eller annet typisk nettverks-tilkoblingspunkt, og innbefatter vanligvis mange eller alle av elementene beskrevet ovenfor relatert til datamaskin 110, selv om bare en minnelag-
 ringsanordning 181 er blitt illustrert i fig. 1. De logiske forbindelsene avbildet innbe-
 10 fatter et lokalt nett (LAN) 171 og et bredbåndet nett (WAN) 173, men kan også innbefatte andre nettverk. Slike nettverksmiljøer er vanlig på kontorer, foretaks-
 datanettverk, interne nettverk (intranett) og internett.

Når brukt i et LAN nettverks-miljø, er datamaskinen 110 koblet til LAN 171 gjennom et nettverksgrensesnitt eller adapter 170. Når brukt i et WAN nettverks-
 15 miljø, innbefatter datamaskinen 110 typisk et modem 172 eller andre hjelpemidler for å oppnå kommunikasjon over WAN 173, slik som internett. Modemet 172, som kan være internt eller eksternt, kan kobles til systembussen 121 via
 brukerinngangs-grensesnitt 160, eller annen passende mekanisme. I et nettverks-
 miljø kan programmoduler avbildet relatert til datamaskinen 110, eller deler derav,
 20 bli lagret i den fjerntliggende minnelagingsenhet. Som et eksempel, og ikke med begrensning, illustrerer fig. 1 fjerntliggende applikasjonsprogrammer 185 som be-
 finner seg på minneanordning 181. Det skal bemerkes at nettverksforbindelsene som er vist er ment som eksempler, og andre måter å oppnå en kommunikasjons-
 forbindelse mellom datamaskinene på, kan brukes.

25

EKSEMPLER PÅ DISTRIBUERTE DATAMASKIN-RAMMEVERK ELLER ARKITEKTURER

Forskjellige distribuerte datamaskin-rammeverk er blitt og blir utviklet i lys av sammenløpingen av personlig databruk og internett. Enkeltindivider og likedan
 30 foretaksbrukere er utstyrt med en tilsynelatende inter-operativt og web-tilrettelagt grensesnitt for applikasjoner og dataanordninger, som gjør data-aktiviteter mer nettleser- eller nettverks-orientert.

For eksempel, MICROSOFT® .NET-plattform innbefatter tjenere, bygge-
 blokkjenester, slik som web-basert datalagring, og nedlastbar anordnings-

programvare. Generelt sagt, tilveiebringer .NET -plattformen (1) muligheten for å gjøre at hele spekteret av data-anordninger kan fungere sammen og gjøre at brukerinformasjon automatisk blir oppdatert og synkronisert på dem alle, (2) økte interaktive muligheter for web-steder, gjort mulig ved mer bruk av XML istedenfor HTML, (3) online-tjenester som har trekk som tilpasset aksess og levering av varer og tjenester til brukeren fra et sentralt sted for styring av forskjellige applikasjoner, slik som e-post, for eksempel, eller programvare, slik som Office .NET, (4) sentralisert datalager, som vil øke effektiviteten og lette tilgang til informasjon, i tillegg til synkronisering av informasjon blant brukere og anordninger, (5) muligheten til å integrere forskjellige kommunikasjonsmedier, slik som e-post, fakser, og telefoner, (6) for utviklere, muligheten til å utvikle arvede moduler, for dermed å øke produktiviteten og redusere antallet programmeringsfeil, og (7) også mange andre kryss-plattform-integrasjonstrekk.

Siden utførelseseksempler her er beskrevet i forbindelse med programvare som befinner seg på en data-anordning, kan også en eller flere deler av oppfinnelsen bli implementert via et operativ-system, API, eller et "mellom-mann"-objekt mellom en co-prosessor og forespørsels-objekt, slik at tjenester kan utføres av, bli støttet av, eller aksessert av alle .NET -språk og tjenester, og også i andre distribuerte data-rammeverk.

20

UTFØRELSESEKSEMPLER

Når en elektronisk fil er blitt formet til mer enn en versjon, i henhold til oppfinnelsen, kan et applikasjons-programgrensesnitt bli konfigurert for å muliggjøre at en programvareapplikasjon assosiert med filen kan kontrollere en prosess for synkronisering av flertallet med versjoner. I en utførelsesform, kan applikasjonen starte synkroniseringen automatisk, mens i en annen utførelsesform, kan applikasjonen starte synkroniseringen basert på bruker-inndata. I en annen utførelsesform, presenterer applikasjonen synkroniseringsvalg tilveiebrakt av applikasjonsfunksjonalitet for brukeren, og tillater brukeren å velge alternativer og utføre oppgaver som bruker applikasjonens funksjonalitet. I en ytterligere utførelsesform, starter en første applikasjon synkroniseringen, mens en andre applikasjon presenterer synkroniseringsvalgene til en bruker.

30

Med referanse nå til fig. 2, omfatter en første datamaskin 210 en første prosessor 220 i operativ kommunikasjon med en lagringsanordning 218. Første data-

maskin 210 kan være en bordmodell for generell bruk eller bærbar datamaskin, en arbeidsstasjon, en spesialisert data-anordning og/eller liknende. På liknende måte kan første prosessor 220 være en prosessor for generell bruk, en spesialisert prosessor, et nettverk av prosessorer eller liknende. Operativt assosiert med første datamaskin 210 er visningsanordning 230. Visningsanordningen 230 kan være
5 enhver innretning for å presentere valg og liknende til en bruker, slik som for eksempel en monitor, høyttaler eller liknende. Slik operativ assosiering av visningsanordning 230 til første datamaskin 210 kan gjøres av ethvert koblet eller trådløst koblet hjelpemiddel. Siden visningsanordning 230 er vist til å være innenfor første datamaskin 210 i fig. 2, kan visningsanordning 230 være lokalisert på ethvert sted
10 som er i operativ kommunikasjon med første prosessor 220, å innbefatte en fjerntliggende lokasjon eller liknende.

Lagringsanordning 218 kan være en harddisk, CD-ROM, magnetisk disk, DVD, RAM og/eller liknende. Som visningsanordning 230, kan lagringsanordning 218 være lokalisert ethvert sted som er i operativ kommunikasjon med første prosessor 220. Operativ kommunikasjon i en slik situasjon kan være koblede, trådløse eller kombinasjoner derav. Inneholdt i lagringsanordning 218 er en første versjon 212 av en elektronisk fil. En slik første versjon 212 kan være enhver type elektronisk fil, slik som for eksempel et tekstbehandlingsdokument, regneark, data-
15 base og/eller liknende.

Første applikasjon 208 og andre applikasjon 214 kan lagres i lagringsanordning 218. Første applikasjon 208 og andre applikasjon 214 kan være en hvilken som helst type av datamaskin-eksekverbare instruksjoner, slik som for eksempel en tekstbehandler, regneark, database, programmert skript, spesialisert programvare eller liknende. I utførelsesformer hvor både første applikasjon 208 og andre applikasjon 214 er tilstede, som diskutert under, behøver bare andre applikasjon 214 være kompatibel med den elektroniske fil. Alternativt, og som også vil bli diskutert nedenfor, kan første applikasjon 208 og andre applikasjon 214 være samme applikasjon. En eller begge av første applikasjon 208 og andre applikasjon
25 214 kan også være lokalisert utenfor lagringsanordning 218, siden bare en av applikasjonene trenger å være i operativ kommunikasjon med første datamaskin 210.

Applikasjons-programgrensesnittet 216 er enhver type datamaskin-eksekverbare instruksjoner tilpasset for å sjekke og sammenlikne to eller flere kopier av en elektronisk fil for synkroniseringsformål, og kan være enkel program-

vare som er spesielt utviklet for synkronisering av to eller flere av et elektronisk dokument, et programvareverktøy som opererer som en komponent av en annen applikasjon, en integrert komponent av første eller andre applikasjon 208, 214 og/eller liknende. Applikasjons-programgrensesnitt 216 kan også utføre andre oppgaver som kan være eller ikke være relatert til synkronisering. Første versjon 212, første applikasjon 208, andre applikasjon 214 og applikasjons-programgrensesnitt 216 kan være innenfor den samme lagringsanordning 218, eller innenfor eller over ethvert antall forskjellige lagringsanordninger 218.

Andre datamaskin 222 består av en andre prosessor 224 og lagringsanordning 228 som inneholder en andre versjon 226 av den elektroniske fil diskutert ovenfor i forbindelse med første versjon 212. Som diskutert ovenfor, kan lagringsanordning 228 være lokalisert på ethvert sted som er i operativ kommunikasjon med andre datamaskin 222, for å inkludere en fjerntliggende lokasjon eller liknende. Lagringsanordning 228 og lagringsanordning 218 kan også, i en utførelsesform, være den samme lagringsanordning. Nettverk 218 forbinder operativt første og andre datamaskin, henholdsvis 210 og 222. Nettverk 218 kan være ethvert kommunikasjonsnettverk tilpasset for slik operativ kommunikasjon, slik som for eksempel et LAN, WAN, trådløst nettverk eller liknende. Som det skal forstås, kan et hvilket som helst antall av første datamaskiner 210 og andre datamaskiner 222 bli operativt koblet til nettverk 218.

Med henvisning nå til fig. 3, starter en synkroniseringsprosess ved trinn 311. I en utførelsesform starter første applikasjon 208 synkroniseringsprosessen, og andre applikasjon 214 fullfører synkroniseringsprosessen. I en annen utførelsesform, er første applikasjon 208 og andre applikasjon 214 en enkelt applikasjon, i hvilket tilfelle den enkle applikasjonen både starter og fullfører synkroniseringen. Mens diskusjonen av fig. 3, som følger, beskriver en prosess som gjelder både første applikasjon 208 og andre applikasjon 214, skal det bemerkes at en enkel applikasjon alene kan utføre alle trinn som er utført av første applikasjon 208 og andre applikasjon 214.

Enhver situasjon kan forårsake at første applikasjon 208 starter en synkroniseringsprosess, slik som for eksempel: åpning, lukking, lagring eller liknende av første versjon 212; ved gjenkjenning av en første og andre versjon av den elektroniske fil; manuell starting ved en brukers forespørsel; starte ved bestemte tidsin-

tervaller; ved at en bruker logger på et system og/eller liknende. Første applikasjon 208 kan i tillegg pålegge enhver tilleggsbetingelse, eller kan bruke erstatnings-betingelser slik som, for eksempel, bare å starte en synkroniseringshendelse når et bestemt antall versjoner eksisterer. Flere detaljer med hensyn til
5 starting av en synkroniseringshendelse følger nedenfor i forbindelse med fig. 4. For å starte prosessen, forårsaker første applikasjon 208 at applikasjons-programgrensesnittet 216 aksesserer første versjon 212 og andre versjon 226. Aksessering av første versjon 212 og andre versjon 226 kan utføres av et hvilket som helst data-hjelpemiddel, slik som ved å se på informasjon assosiert med hver
10 versjon, finne hver versjon ved bruk av søkings-funksjonalitet, lokalisere hver versjon ved å bruke identifiserende informasjon tilveiebrakt av første applikasjon eller liknende.

Første applikasjon 208 kan utstyre grensesnitt 216 med en hvilken som helst mengde av identifiserende informasjon om andre versjon 226, slik som for
15 eksempel et filnavn, lokasjon, adresse og/eller liknende. Grensesnittet 216 kan tilpasses til aktivt å lokalisere andre versjon 226, eller kan ha begrenset funksjonalitet som bare tillater visning av en lokasjon som spesifisert av første applikasjon 208, første versjon 212 eller liknende. Grensesnittet 216 kan også bli konfigurert til å utføre ethvert antall av tilleggsoppgaver som til og med kan være ubeslektet
20 med synkroniseringsprosessen. Ved trinn 313 lokaliserer applikasjons-programgrensesnittet 216 andre versjon 226 og sjekker synkroniseringsstatus. Applikasjons-programgrensesnittet 216 avgjør status ved å sammenlikne informasjon, slik som for eksempel tid og/eller dato ved siste oppdatering eller liknende, for første versjon 212 mot andre versjon 226. Synkroniseringsstatus kan
25 også avgjøres ved ethvert kriterium i tillegg til eller erstattet med informasjonen nevnt ovenfor. Hvis informasjonen stemmer overens med informasjon og/eller kriterier spesifisert av første applikasjon 208, har første og andre versjon 212, 226, status som synkronisert. Hvis informasjonen ikke stemmer overens, har ikke første og andre versjon 212, 226 status som synkronisert, siden en slik uoverensstem-
30 melse indikerer muligheten for at endringer er blitt gjort på en av slike versjoner 212, 226. Alternativt, hvis applikasjons-programgrensesnittet 216 finner en feil, en tilstand det ikke er sørget for, en tilstand som første applikasjon 208 har spesifisert som grunn for avbrytelse av synkroniseringen, eller liknende, fortsetter grensesnitt 216 til trinn 330. Ved trinn 330 informerer grensesnitt 216 første applikasjon 208

med betingelsen, og første applikasjon fortsetter følgelig, som kan inkludere enhver type handling eller ikke-handling.

Ved trinn 315, kan applikasjons-programgrensesnittet 216 ta en av to handlingsveier, avhengig av resultatet av synkroniserings-bestemmelsen i trinn 313.

5 Hvis applikasjon-programgrensesnittet har avgjort at første versjon 212 og andre versjon 226 er synkronisert, fortsetter grensesnittet 216 til trinn 325. Ved trinn 325 rapporterer applikasjons-programgrensesnittet til første applikasjon 208 at første versjon 212 og andre versjon 226 er synkronisert. En slik rapport indikerer til første applikasjon 208 at det på nåværende tidspunkt ikke er nødvendig med videre syn-

10 kroniserings-aktiviteter, siden første versjon 212 omfatter den siste versjon av den elektroniske fil. Rapporten kan også indikere at en bruker kan anvende første applikasjon 208 eller andre versjon 214 for å endre første versjon 212.

Ved trinn 323 avsluttes synkroniseringsprosessen. Ved dette trinnet kan første applikasjon stoppe å fungere, slik som for eksempel hvis en slik synkroniseringsprosess tok sted da første applikasjon 208 lukket første versjon 212 av det

15 elektroniske dokument. I en slik situasjon, etter at synkroniseringen har skjedd, har derfor første applikasjon 208 ingen flere funksjoner å utføre og kan avbrytes manuelt av bruker eller automatisk. I en annen utførelsesform starter første applikasjon 208 synkroniseringsprosessen ved åpning av en første versjon 212. I en slik

20 situasjon vil ikke applikasjon avsluttes ved fullføring av en synkroniseringshendelse, men vil heller tillate bruker å begynne endring av første versjon 212 ved bruk av første applikasjons- 208 funksjonalitet, hvis tilgjengelig.

Hvis imidlertid, ved trinn 315, applikasjons-programgrensesnittet 216 har avgjort at første versjon 212 og andre versjon 226 ikke er synkronisert, fortsetter

25 grensesnitt 216 til trinn 317. Ved trinn 317 rapporterer applikasjons-programgrensesnitt 216 til første applikasjon 208 og/eller andre applikasjon 214 at første versjon 212 og andre versjon 226 ikke er synkronisert. En slik rapport forårsaker at andre applikasjon 214 begynner synkronisering av første versjon 212 og andre versjon 226 med bruker-inndata, som diskutert under i forbindelse med trinn

30 319 og 321.

Ved trinn 319 presenterer andre applikasjon 214 et brukergrensesnitt til en bruker ved å vise brukergrensesnittet på en visningsanordning 230. Et slikt brukergrensesnitt kan være tekst-basert og/eller grafikk-vindu, lyd-spørsmål og/eller liknende, og enhver slik konfigurasjon av et brukergrensesnitt er like gjennomførbare

med en utførelsesform av den foreliggende oppfinnelse. Ved presentasjon av et slikt brukergrensesnitt, kan andre applikasjon 214 presentere ett eller flere valg til en bruker ved bruk av andre applikasjon-funksjonalitet eller liknende. For eksempel, hvis andre applikasjon 214 er en tekstbehandler, kan andre applikasjon vise et

5 synkroniseringsvalg til en bruker innenfor et tekstbehandler-brukermiljø i andre applikasjon 214, slik som for eksempel et redigeringsvindu med en side-visning av en slik første versjon 212 og/eller liknende. Andre applikasjon 214 kan også belyse forskjeller mellom første versjon 212 og andre versjon 226 ved bruk av farge, lyder, grafikk og liknende for bedre å gjøre bruker i stand til å gjøre synkroniseringsvalg. Andre applikasjon 214 kan også tilveiebringe tilleggsinformasjon, slik

10 som for eksempel identiteten til hver bruker som endret en av versjon 212, 226, og kan også tillate en bruker å endre en versjon 212, 226 hvis en bruker har rettigheter, i form av en sikkerhetsklarering eller liknende, til å gjøre dette. Andre applikasjon 214 kan gjøre det mulig for bruker å, for eksempel, blande versjon 212,

15 226, i en tredje kopi, for så å innlemme endringer i begge versjoner 212, 226. Et tilleggsvalg kan også være i form av rettledning utledet fra andre applikasjons 214 funksjonalitet, slik som for eksempel en anbefalt handlingsvei og/eller liknende. Som det skal bemerkes, kan et valg være ethvert valg som kan fremskaffes av andre applikasjon 214, ved bruk av andre applikasjons 214 funksjonalitet med

20 hensyn til første versjon 212 og andre versjon 226.

Ved trinn 321, mottar og behandler andre applikasjon 214 bruker-inndata. Slike bruker-inndata kan omfatte enhver mengde informasjon angående slik synkronisering. For eksempel kan bruker-inndata velge ut en bestemt endring av enten første versjon 212 eller andre versjon 226 av den elektroniske fil som brukeren

25 ønsker å beholde. I en annen utførelsesform kan bruker-inndata omfatte et valg av enten første versjon 212 eller andre versjon 226 som bruker ønsker å beholde i sin helhet. I ytterligere en utførelsesform kan bruker-inndata omfatte redigering av første og/eller andre versjon 212, 226 ved bruk av andre applikasjons 214 funksjonalitet som, ved fullføring, resulterer i synkronisering av første versjon 212 og

30 andre versjon 226. Som det skal bemerkes, kan bruker-inndata omfatte enhver inndata gjort tilgjengelig av andre applikasjon 214 for valg eller bruk av en bruker, og alle slike inn-data er gjennomførbare med en utførelsesform ifølge den foreliggende oppfinnelse.

Som det også skal bemerkes, kan trinn 319 og 321 repetere, siden slike brukergrensesnitt kan vises flere ganger og/eller oppdateres avhengig av bruker-inndata. For eksempel kan brukergrensesnittet bli konfigurert på nytt av andre applikasjon 214 basert på bruker-inndata. En slik rekonfigurering kan skje hvis, for
5 eksempel, bruker velger en av et flertall forskjeller mellom første versjon 212 og andre versjon 226, og andre applikasjon 214 viser brukergrensesnitt på nytt til bruker for å reflektere et slikt valg. Trinn 319 og 321 kan repetere et hvilket som helst antall ganger, eller ikke i det hele tatt, og enhver slik situasjon er like gjennomførbar med en utførelsesform ifølge den foreliggende oppfinnelse. Andre applikasjon 214 kan også rapportere ethvert resultat av synkroniseringen til første
10 applikasjon 208 eller til grensesnitt 216. Resultatene rapportert av andre applikasjon 214 kan være en hvilken som helst informasjon assosiert med synkroniseringen, slik som valgselektering(er), synkroniseringsresultat(er) eller liknende.

Andre applikasjon 214 kan fortsette til trinn 323, slutten på synkroniseringen, når en triggings-hendelse inntreffer. En slik triggings-hendelse kan skje på et
15 hvilket som helst tidspunkt, slik som for eksempel når bruker har fullført foreningen av alle forskjeller mellom første versjon 212 og andre versjon 226, når bruker leder den andre applikasjon 214 til å avbryte eller avslutte synkroniseringsprosessen uten å fullføre synkroniseringen, på en forhåndbestemt tid, eller liknende. Ved trinn
20 323 avsluttes synkroniseringsprosessen. Første applikasjon 208 eller andre applikasjon 214 kan da, som nevnt ovenfor i forbindelse med trinn 325, utføre enhver handling ved trinn 323 som er i samsvar med sammenhengen hvor synkroniseringen fant sted.

Nå med henvisning til fig. 4, er det vist en detaljert oversikt av trinn 311 i
25 henhold til en utførelsesform av den foreliggende oppfinnelse. Som beskrevet ovenfor i forbindelse med fig. 3, starter første applikasjon 208 i trinn 311 en synkroniseringsprosess. Fig. 4 illustrerer nærmere bestemt en utførelsesform av et slikt oppstartstrinn, hvor første applikasjon 208 starter en slik synkroniseringsprosess ved åpning av første versjon 212 av en elektronisk fil. Som diskutert ovenfor,
30 kan første applikasjon 208 starte en slik synkroniseringsprosess på et hvilket som helst tidspunkt, slik som for eksempel ved åpning, lagring, lukking av en slik første versjon 212, på faste eller variable intervaller, ved bruker-inndata og/eller liknende.

Ved trinn 411, åpner første applikasjon 208 første versjon 212 av en elektronisk fil. Slik åpning vil vanligvis skje ved bruker-inndata til å åpne en slik første

versjon 212, men enhver innretning for initiering av slik åpning av første versjon 212 er gjennomførbar med en utførelsesform ifølge den foreliggende oppfinnelse. Ved trinn 413, bestemmer første applikasjon 208 om første versjon 212 har en assosiert andre versjon 226. Første applikasjon 208 kan fortsette til trinn 413 ved

5 full åpning av første versjon 212 i trinn 411, eller kan fortsette til trinn 413 på et hvilket som helst tidspunkt under en slik åpningsprosess, inkludert, for eksempel, før åpning av første versjon 212 ved trinn 411. Enhver metode kan også benyttes for å gjøre det mulig for første applikasjon 208 å gjøre en slik bestemmelse. For eksempel kan informasjon, slik som et elektronisk merke eller liknende, bli asso-

10 siert med en slik første versjon 212 for å gjøre det mulig for første applikasjon 208 å gjøre disse bestemmelser. I motsetning til dette, kan første applikasjon 208 søke enhver eller alle data-anordninger operativt koblet til nettverk 218, slik som for eksempel andre datamaskin 222, for slik bestemmelse. Som det skal forstås, kan første applikasjon 208 bruke enhver metode for å bestemme eksistensen av andre

15 versjon 226, mens den fortsatt er forenelig med en utførelsesform av den foreliggende oppfinnelse.

Hvis, i trinn 413, første applikasjon 208 ikke finner andre versjon 226, bestemmer første applikasjon 208 at første versjon 212 er den eneste kopi av en slik elektronisk fil og fortsetter til trinn 417. Ved trinn 417 behandler første applikasjon

20 208 første versjon normalt. Slik normal behandling kan være enhver type behandling som første applikasjon 208 typisk utfører på elektroniske filer av samme format som første versjon 212, muligens i henhold til enhver bruker-inndata. For eksempel, i fig. 4, var første applikasjon 208 i ferd med å åpne første versjon 212 da første applikasjon 208 startet en slik synkroniseringsprosess i henhold til trinn 311.

25 Når det ikke er funnet behov for synkronisering, fordi andre versjon 226 ikke er tilgjengelig, fullfører første applikasjon 208 åpningen av første versjon 212 og fortsetter følgelig.

Hvis, i trinn 413, første applikasjon 208 finner andre versjon 226, fortsetter første applikasjon til trinn 415. I trinn 415, kontakter første applikasjon 208

30 applikasjons-programgrensesnitt 216 for å begynne synkroniseringen av første versjon 212 og andre versjon 226. Ved å kontakte applikasjons-programgrensesnitt 216, fortsetter en slik synkroniseringsprosess til trinn 313, som diskutert ovenfor i forbindelse med fig. 3.

På denne måte har fremgangsmåter og apparat for styring av fil-replikering i applikasjoner blitt presentert. Siden den foreliggende oppfinnelse er blitt beskrevet i forbindelse med de foretrukne utførelsesformer av de forskjellige figurer, skal det forstås at andre liknende utførelsesformer kan brukes, eller modifikasjoner og tillegg kan gjøres på den beskrevne utførelsesform for å utføre den samme funksjon som den foreliggende oppfinnelse, uten å avvike fra denne. For eksempel vil en som er kjent på fagområdet erkjenne at den foreliggende oppfinnelse som er beskrevet i den foreliggende søknad kan gjelde enhver data-anordning eller miljø, uansett koblet eller trådløst, og kan anvendes på ethvert antall av slike data-anordninger koblet via et kommunikasjons-nettverk og som kommuniserer over nettverket. Det skal videre understrekes at en varietet av datamaskin-plattformer, innbefattet operativsystem for håndholdte anordninger og andre applikasjons-spesifikke operativsystemer, er overveid, spesielt etter at antallet trådløse nettverkskoblede anordninger fortsetter å bre seg. Den foreliggende oppfinnelse kan videre implementeres i eller over et flertall prosessor-brikker eller anordninger, og lagring kan på lignende måte virke inn på et flertall anordninger. Derfor skal ikke den foreliggende oppfinnelse være begrenset til en enkelt utførelsesform, men skal heller oppfattes i bredde og omfang i henhold til de vedføyde krav.

PATENTKRAV

1. En fremgangsmåte for synkronisering av en første og en andre versjon av en elektronisk fil, hvor fremgangsmåten omfatter:

5 å starte (311), ved en første programvareapplikasjon som opererer på en første datamaskin, en synkroniserings-hendelse for å synkronisere den første versjonen med den andre versjonen;

 å forårsake (313), ved den første programvareapplikasjonen at et applikasjons-programgrensesnitt aksesserer nevnte første og andre versjoner;

10 å sammenlikne (315), ved applikasjonsprogramgrensesnittet, informasjon assosiert med hver av de nevnte første og andre versjoner for å bestemme hvorvidt den første og den andre versjonen er synkronisert, og hvis ikke:

 å rapportere (317), ved applikasjonsprogramgrensesnittet, til den første programvareapplikasjonen at den første og den andre versjonen ikke er synkroni-
15 sert;

 å forårsake at en andre programvareapplikasjon frembringer et brukergrensesnitt i henhold til et trekk i den andre programvareapplikasjonen, hvor den elektroniske fil er i samsvar med den andre programvareapplikasjon; og

 å vise (319) brukergrensesnittet som presenterer én eller flere synkroniseringsmuligheter for å synkronisere den første og den andre versjonen på en visningsanordning;

 å motta (321), ved den andre programvareapplikasjonen, et valg av én av de én eller flere synkroniseringsmuligheter fra en bruker via brukergrensesnittet; og

25 å synkronisere første og andre versjoner i henhold til den valgte synkroniseringsmulighet.

2. Fremgangsmåte ifølge krav 1, hvor den første og andre programvareapplikasjon er den samme applikasjonen.

30

3. Fremgangsmåte ifølge krav 1 eller 2, hvor den første versjon er lagret på en første datamaskin, hvor den andre versjon er lagret på en andre datamaskin, og nevnte første og andre datamaskin er operativt koblet sammen av et kommunikasjonsnettverk.

4. Fremgangsmåte ifølge et hvilket som helst av kravene 1 til 3, hvor trinnet med å synkronisere den første og andre versjon omfatter å overskrive den andre versjon av den elektroniske fil med den andre versjon av den elektroniske fil.

5

5. Fremgangsmåte ifølge krav 4, hvor trinnet med å synkronisere den første og andre versjon omfatter å overskrive den første og andre versjon med en tredje versjon av den elektroniske fil, nevnte tredje versjon omfatter en del av den første versjon av den elektroniske fil.

10

6. Fremgangsmåte ifølge et hvilket som helst av kravene 1 til 3, hvor trinnet med å synkronisere den første og andre versjon omfatter å overskrive den første og andre versjon med en tredje versjon av den elektroniske fil, nevnte tredje versjon omfatter en del av den andre versjon av den elektroniske fil.

15

7. Fremgangsmåte ifølge et hvilket som helst av kravene 1 til 3, hvor trinnet med å synkronisere den første og andre versjon omfatter å blande den første og andre versjon til en tredje versjon i henhold til inntastninger mottatt fra brukergrensesnittet og å overskrive den første og andre versjon med den tredje versjon.

20

8. Fremgangsmåte ifølge krav 1, som videre omfatter:
å åpne den første og andre versjon ved bruk av den andre programvareapplikasjon; og
å vise den første og andre versjonen i brukergrensesnittet.

25

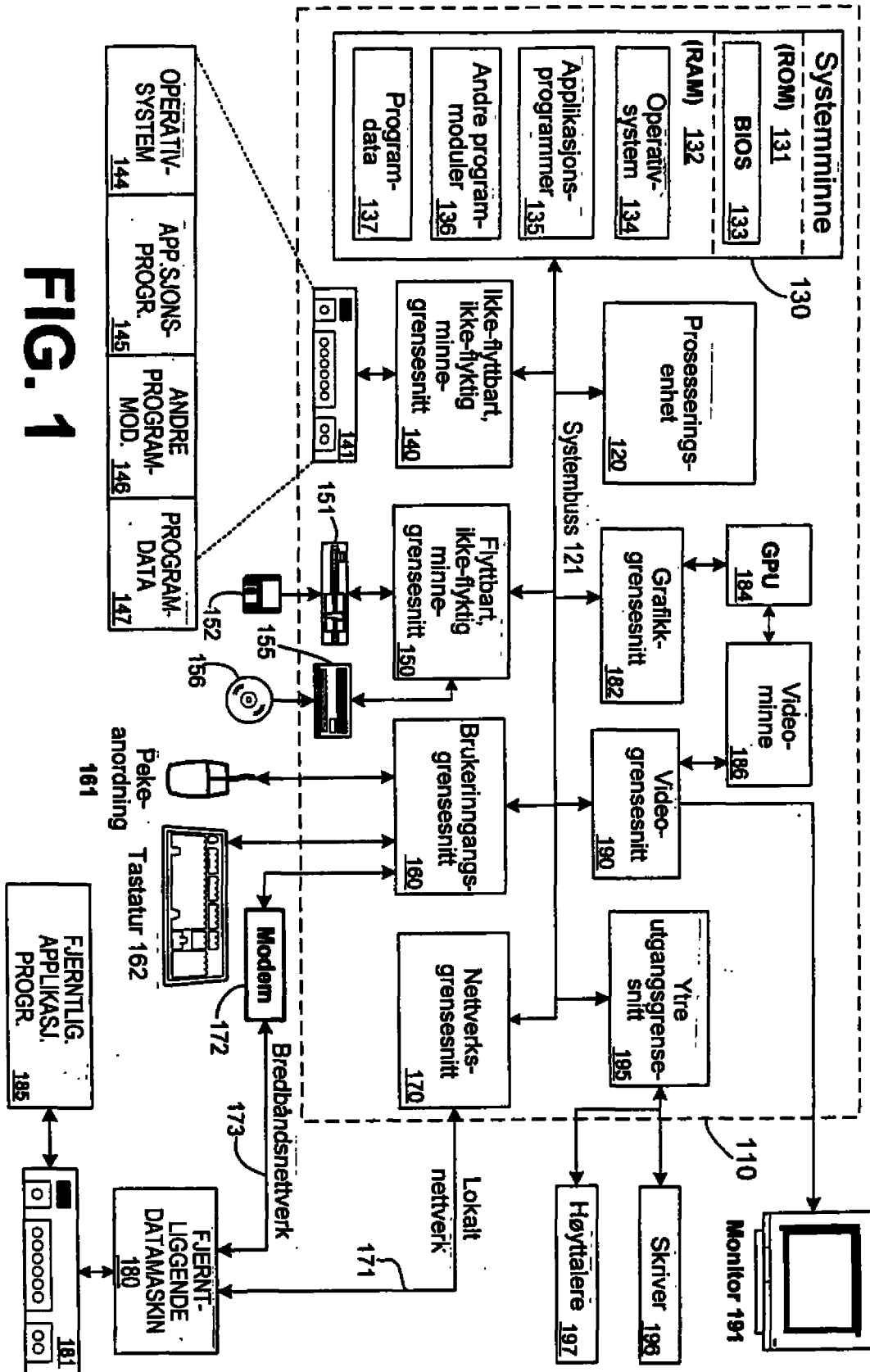
9. Fremgangsmåte ifølge krav 8, hvor trinnet med synkronisering av den første og andre versjon omfatter å overskrive den første og andre versjon i henhold til bruker-data.

30

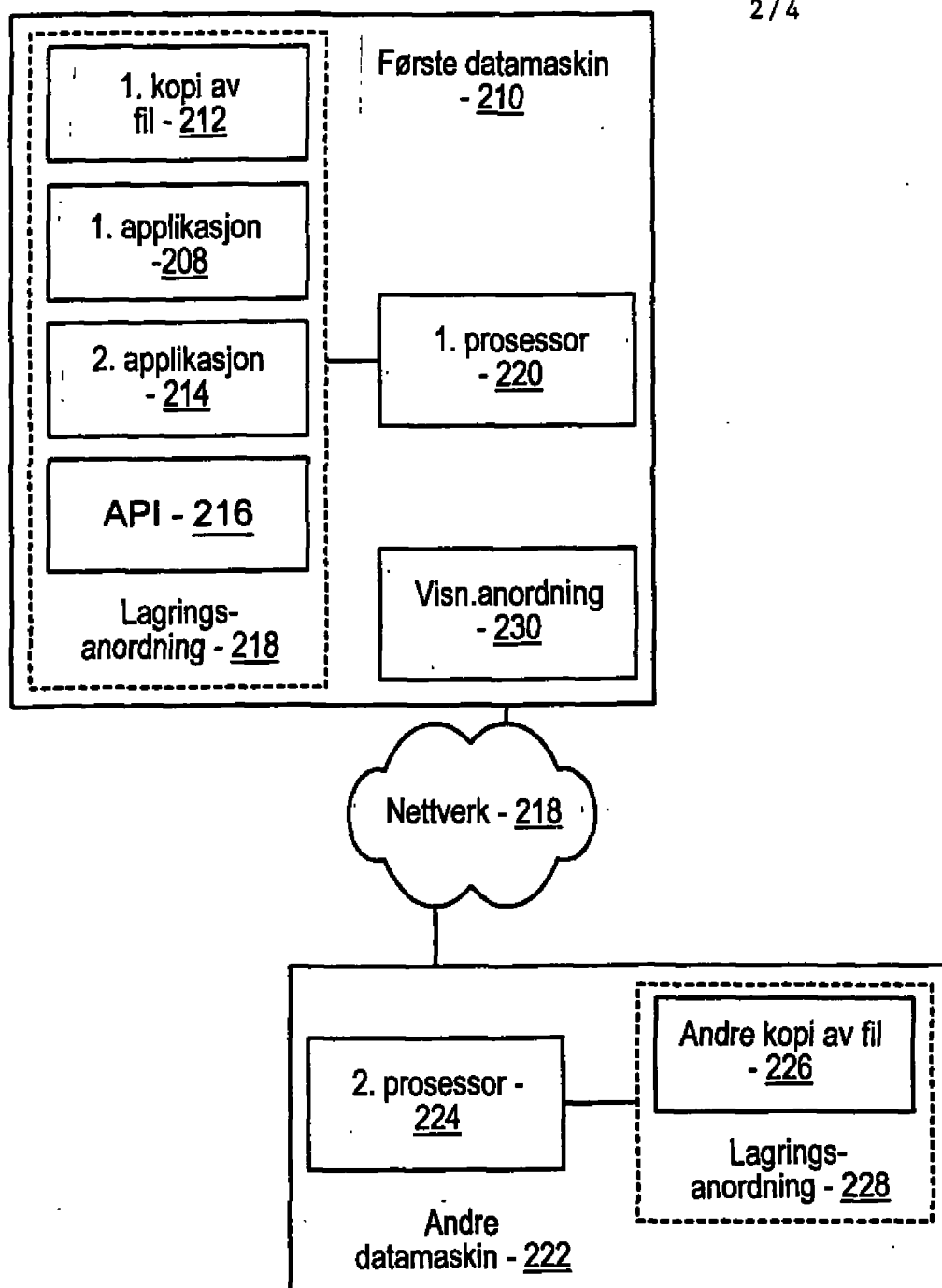
10. Fremgangsmåte ifølge ett av kravene 1 til 9, hvor trinnet med å starte (311) synkroniserings-hendelsen skjer når den første programvareapplikasjon finner informasjon i samsvar med den første versjon av den elektroniske fil som tyder på den andre versjon av den elektroniske fil.

11. Fremgangsmåte ifølge ett av kravene 1 til 9, hvor trinnet med å starte (311) synkroniserings-hendelsen skjer når den første programvareapplikasjon åpner den første versjon av den elektroniske fil.
- 5 12. Fremgangsmåte ifølge ett av kravene 1 til 9, hvor trinnet med å starte (311) synkroniserings-hendelsen skjer når den første programvareapplikasjon lagrer den første versjon av den elektroniske fil til en lagringsanordning på den første datamaskin.
- 10 13. Fremgangsmåte ifølge ett av kravene 1 til 9, hvor trinnet med å starte (311) synkroniserings-hendelsen skjer når den første programvareapplikasjon gjenkjenner den første versjon og andre versjon.
- 15 14. Et datamaskin-lesbart medium som har datamaskin-eksekverbare instruksjoner lagret derpå som når de utføres av en eller flere datamaskiner forårsaker den ene eller flere datamaskinen å utføre en fremgangsmåte i henhold til ett av kravene 1 til 13.

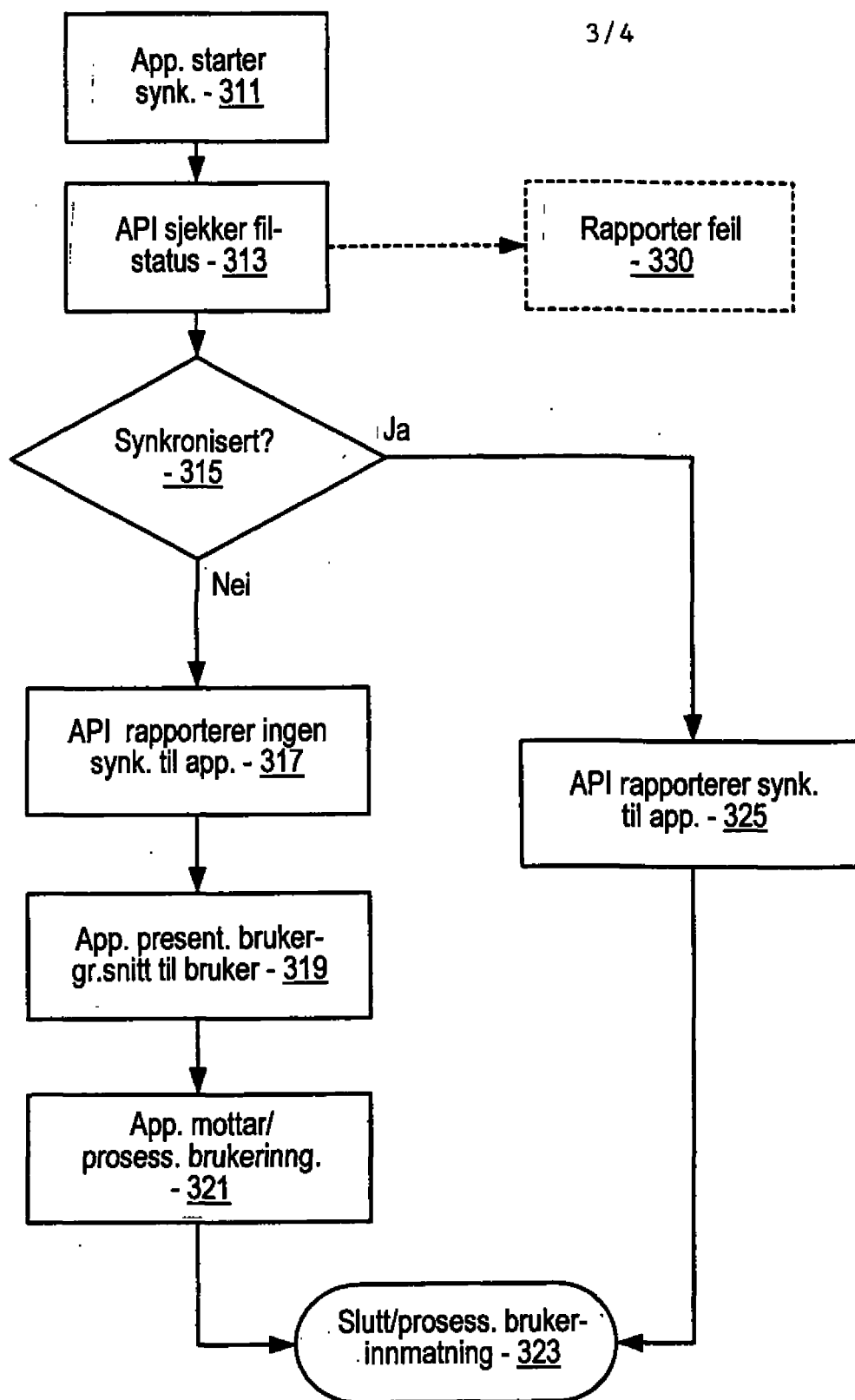
1/4

Data-miljø 100**FIG. 1**

2 / 4

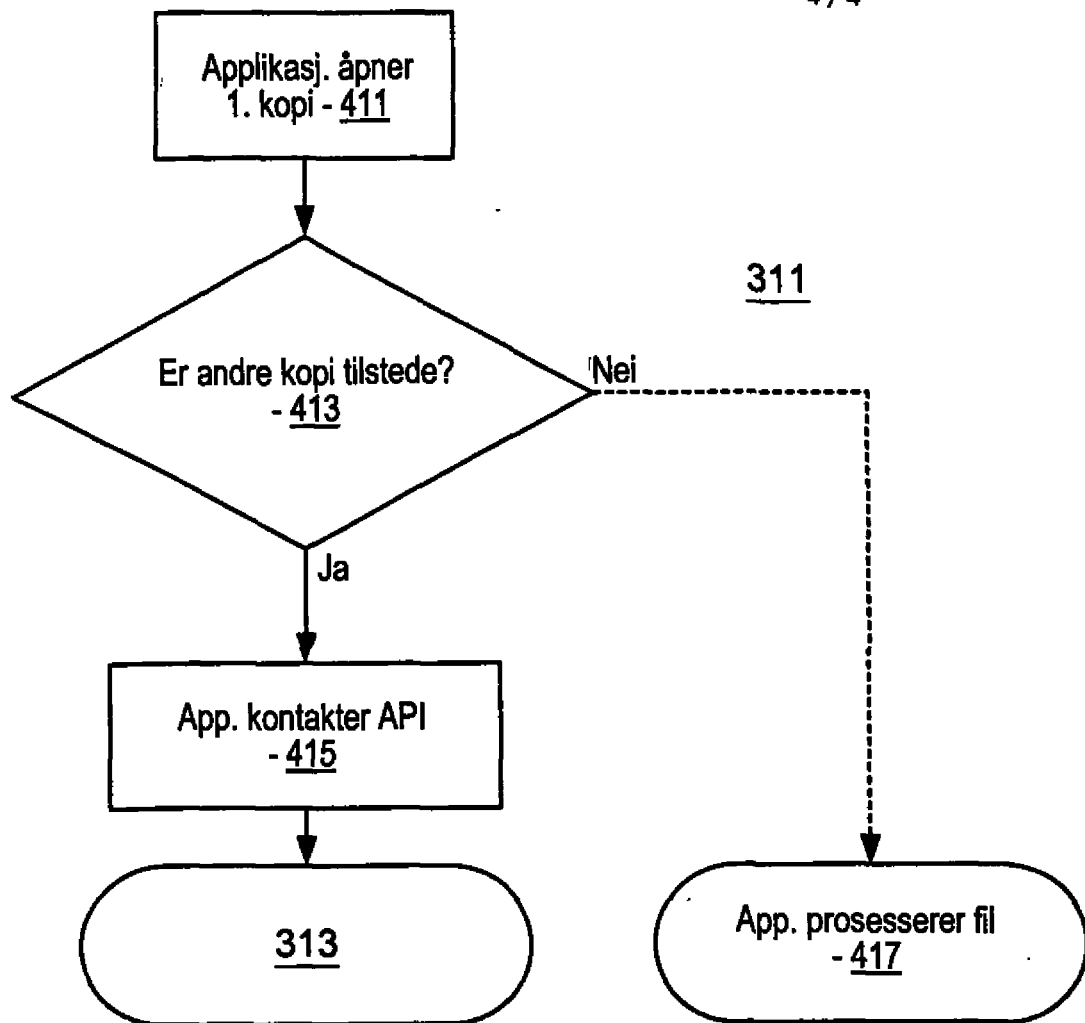


FIGUR 2



FIGUR 3

4 / 4



FIGUR 4