

R-RCT

Nationell rekommendation R-RCT

Teknisk och funktionell översikt

Innehållsförteckning

1. Introduktion	3
1.1. Syfte	3
1.2. Definitioner och förkortningar	3
1.3. Referenser	4
2. Vad gör R-RCT-ramverket?	4
3. Ett ramverk, inte en färdig lösning	4
3.1. Ett första exempel.....	4
4. Systemöversikt, utifrån och in	8
4.1. R-RCT-implementationen i sin helhet	8
4.2. De olika delsystemen.....	9
4.2.1. Externt EDC.....	10
4.3. Kvalitetsregistret.....	10
4.3.1. Randomisering och rapportering av studiedata	11
4.4. Studiens webbapplikation	14
5. Indataformat	15
5.1. Metadata i CDISC ODM-format.....	15
5.2. Variabelns id återspeglar hierarkin	15
5.3. JSON-struktur i REST-anropet för lagring av data.....	15
5.3.1. Ordningsnummer för repeterande komponenter	16
6. Arkitekturens mål och begränsningar	16
6.1. Modularitet.....	16
6.2. Testbarhet.....	16
6.3. Begränsningar.....	16
7. Kvalitet	17
7.1. Modularitet.....	17

1. Introduktion

1.1. Syfte

Dokumentet ska ge en funktionell och teknisk översiktsbild av R-RCT-ramverket utan att gå in på tekniska detaljer. Förmodade läsare inkluderar därmed roller som IT-strateg, projektledare eller liknande. Även för en utvecklare ska dokumentet ge en bra sammanfattning, med mer tekniskt fördjupning i dokumentet [UTV_DOC]. Läsaren förutses ha viss kunskap om kvalitetsregister och även viss kunskap om kliniska studier. Förhoppningen är att läsaren efteråt ska se hur R-RCT passar in i den egna organisationen, samt ha fått en allmän teknisk förståelse av ramverket.

1.2. Definitioner och förkortningar

Definition	Förklaring
R-RCT	Registry-based Randomised Clinical Trial – en randomiserad klinisk studie som genomförs genom att screening, rekrytering och inrapportering av studiedata genomförs baserat på den information som tillhandahålls av ett kvalitetsregister.
Kvalitetsregister	Datorstöd för lagring och analys av data som kan användas för kvalitetsförbättring inom sjukvården.
Studie-webbapplikation	En webbapplikation som använder R-RCT-ramverket för att implementera datorstöd för en given R-RCT.
R-RCT-release	En uppsättning jar-filer och dokumentation som hör till en viss version av R-RCT-ramverket.
Artefakt	En komponent i form av en fysisk fil, till exempel en jar-fil eller ett dokument.
jar-fil	Kompilerad Java-kod och konfigurationsfiler, paketerade i formatet Java ARchive (JAR).
war-fil	En webbapplikation, paketerad i formatet Web ARchive (WAR).
Investigator/prövare	Prövare är en läkare på klinik med rätt att randomiseras och föra in studiedata. Det engelska "investigator" används ofta i dagligt tal istället för prövare
PM	Projekt Manager, projektledare
DM	Data Manager
Subject	Det engelska ordet subject används här i svensk text för att representera en person som är rekryterad till studien.
property	Ett sätt att peka ut en del av systemet som inte ska vara kompillerad kod. För att till exempel ange adressen till en extern server kan den kompilerade koden använda nyckeln <code>clinicaltrials.edc.url</code> , och värdet för nyckeln skrivs i en textfil som läses upp av systemet i runtime: <code>clinicaltrials.edc.url=https://edcs.r.us/mystudy</code> .
classpath	Den uppsättning av referenser som en Java-applikation använder för att hitta klasser, textfiler, bilder och andra resurser. En classpath består vanligtvis av en rad kataloger i ett filsystem, till exempel <code>"/opt/classes:/opt/images:/lib/java"</code> .
runtime	Betecknar den tidpunkt då ett mjukvarusystem körs, i kontrast till tiden då systemet kompileras och byggs.
REST	En arkitektur för maskin till maskin-kommunikation.
EDC	Electronic Data Capture, datoriserat system för insamling och hantering av data i kliniska studier.

JSON	JavaScriptObjectNotation, en formatspecifikation för att representera data och datastrukturer vars främsta användning är i kommunikation mellan system på internet.
CDISC	(The) Clinical Data Interchange Standards Consortium, är en organisation som bildades 1997 och som syftar till att etablera standarder för data inom forskning och sjukvård.
CDISC ODM	CDISC Operational Data Model, den standard som CDISC utarbetat för kommunikation mellan system inom klinisk forskning. Standarden är leverantörsberoende och inkluderar en plattformsnöutral standard för lagring av data från kliniska studier.

1.3. Referenser

Referens	Dokument
C4MODEL	http://c4model.com/
SPRING	https://projects.spring.io/spring-framework/
UTV_DOC	Dokumentet "Nationella riktlinjer – Bilaga Utvecklardokumentation"

2. Vad gör R-RCT-ramverket?

I en R-RCT vill man att provare ska screena och randomisera patienter samt mata in studiedata via det kvalitetsregister man redan använder dagligen. För att uppnå detta kan man låta registret använda webbtjänster för de funktioner en provare använder.

Webbtjänsterna ska sköta hantering av Screening Log, Subject Log, randomisering och lagring av studiedata. R-RCT-ramverket innehåller mjukvarukomponenter som gör det enkelt att bygga sådana webbtjänster.

3. Ett ramverk, inte en färdig lösning

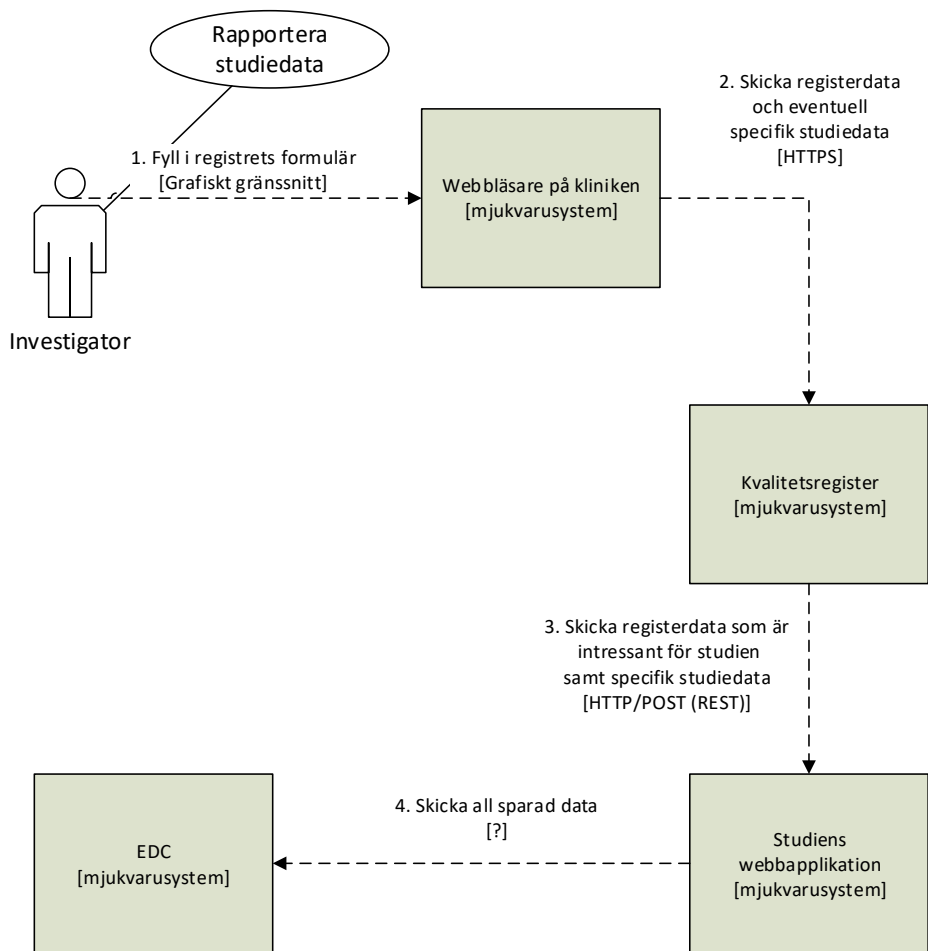
R-RCT-ramverket utgörs av ett antal mjukvarukomponenter som kan användas när man implementerar en R-RCT. Ramverket underlättar screening, randomisering, inmatning av studiedata, samt hantering av Screening Log och Subject Log. För att implementera en R-RCT krävs att ett mjukvarusystem utvecklas som exponerar ramverkets tjänster. Lättast är att skapa en webbapplikation skriven i Java och som använder Spring Framework, samt lagrar studiens metadata och data i en databas som kan användas via Spring- och Hibernate-kopplingar.

3.1. Ett första exempel

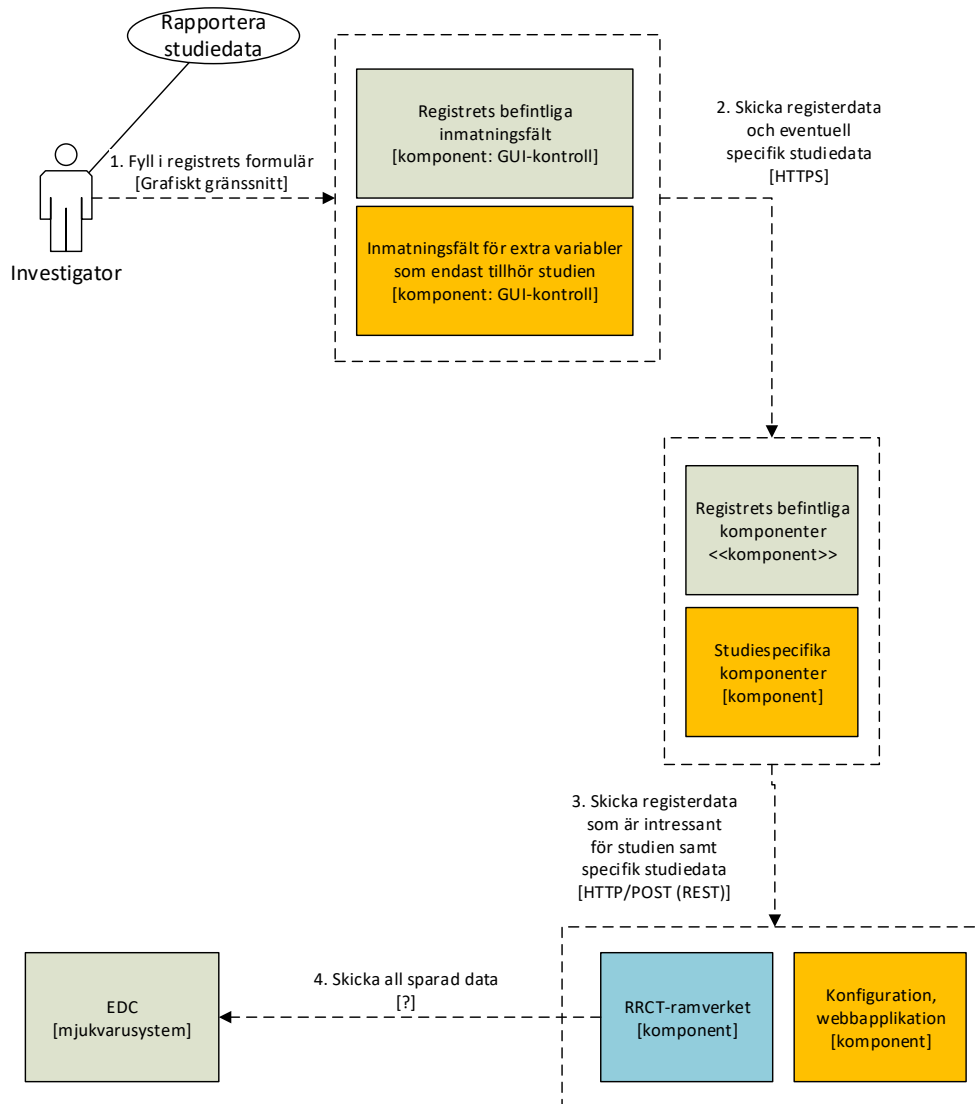
För att få en snabb överblick över ramverkets roll i en R-RCT-implementation ges här två bilder av hur användningsfallet Rapportera studiedata löper genom alla delsystem, på containernivå samt på komponentnivå [C4MODEL]. Den senare bilden belyser att R-RCT-ramverket bara är en liten del av hela implementationen. Utvecklarna måste själva sätta upp en webbapplikation så att REST-tjänsterna i ramverket exponeras och så att kommunikation mot databas och en eventuell EDC kan upprättas. Utvecklarna måste också själva implementera studiespecifik logik och inmatningsfält i registret.

Kommunikationsprotokollet mellan studiens webbapplikation och EDC är markerat med ett frågetecken eftersom olika EDC-system använder olika protokoll. För att integrera med ett specifikt EDC, måste EDC-specifik kod skrivas.

Figur 1. Dynamiskt containerdiagram för rapportering av studiedata [C4MODEL].



Figur 2. Dynamiskt komponentdiagram för rapportering av studiedata, med studiespecifik implementation i gult och R-RCT-ramverket i blått [C4MODEL].



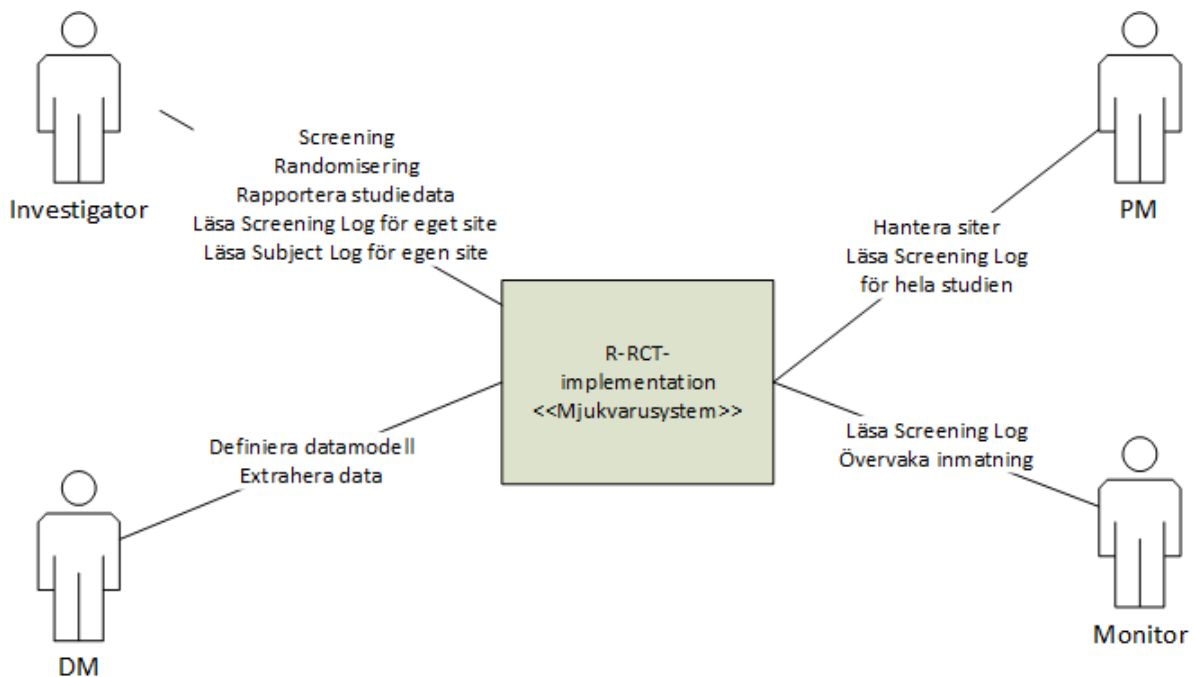
4. Systemöversikt, utifrån och in

För att ge en bild av de funktioner som R-RCT-ramverket fyller i en R-RCT-implementation, ges här en översikt i ett antal diagram. Diagrammen följer C4-modellen, där man börjar med ett fågelperspektiv, för att sedan steg för steg visa en ökad detaljnivå [C4MODEL].

4.1. R-RCT-implementationen i sin helhet

I Figur 3 betraktas hela R-RCT-implementationen som ett enda system. Systemet utgörs av kvalitetsregister, studiens webbapplikation samt ett EDC. Bilden visar hur olika aktörer använder systemet.

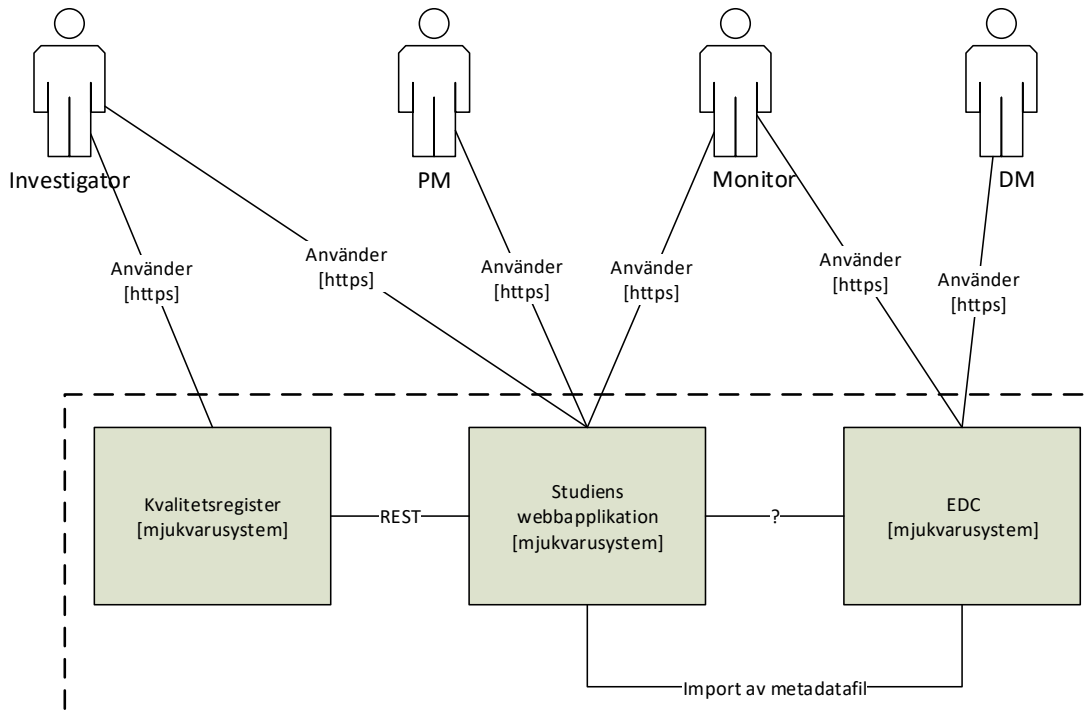
Figur 3. R-RCT-implementationen i sin helhet, med aktörer och användningsfall.



4.2. De olika delsystemen

Figur 4 visar ett exempel på hur R-RCT-implementationen kan delas upp i olika delsystem. Kommunikationsprotokollet mellan studiens webbapplikation och EDC är markerat med ett frågetecken eftersom olika EDC-system använder olika protokoll. Observera också att det måste skrivas EDC-specifik kod för denna integration.

Figur 4. Ett exempel på hur implementationen kan delas upp i tre system.



4.2.1. Externt EDC

Studier som inte kräver att data lagras i ett EDC, kan helt förlita sig på R-RCT-ramverkets interna lagring av data. För studier där ett externt EDC måste kopplas på, erbjuder R-RCT-ramverket integrationspunkter i koden, där kod måste skrivas för att integrera mot aktuell EDC. Detta är mer detaljerat beskrivet i [UTV_DOC].

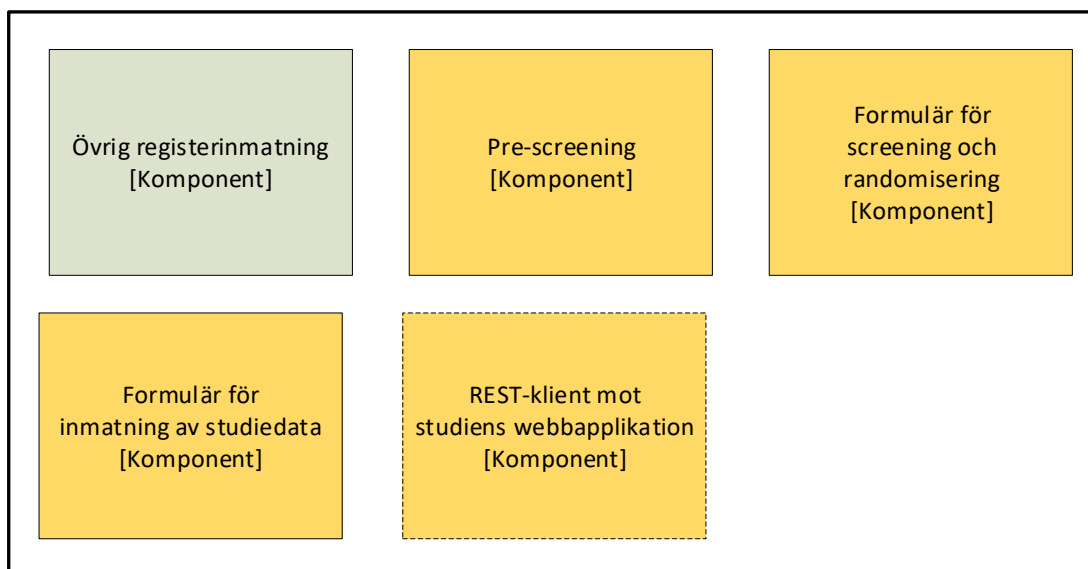
I det fortsatta dokumentet är oftast ett externt EDC förmodat. Om inget externt EDC behövs kan dessa delar förstås ignoreras.

4.3. Kvalitetsregistret

För att implementera en R-RCT krävs utveckling av studiespecifika komponenter i kvalitetsregistret. All klientlogik och grafisk presentation som stödjer prövarens användningsfall måste implementeras i registret. I bilden markeras alla komponenter som måste implementeras i registret med gul färg.

Observera komponenten med streckad kant, i stället för att implementera denna själv kan man använda färdiga Java-klienter som medföljer ramverket.

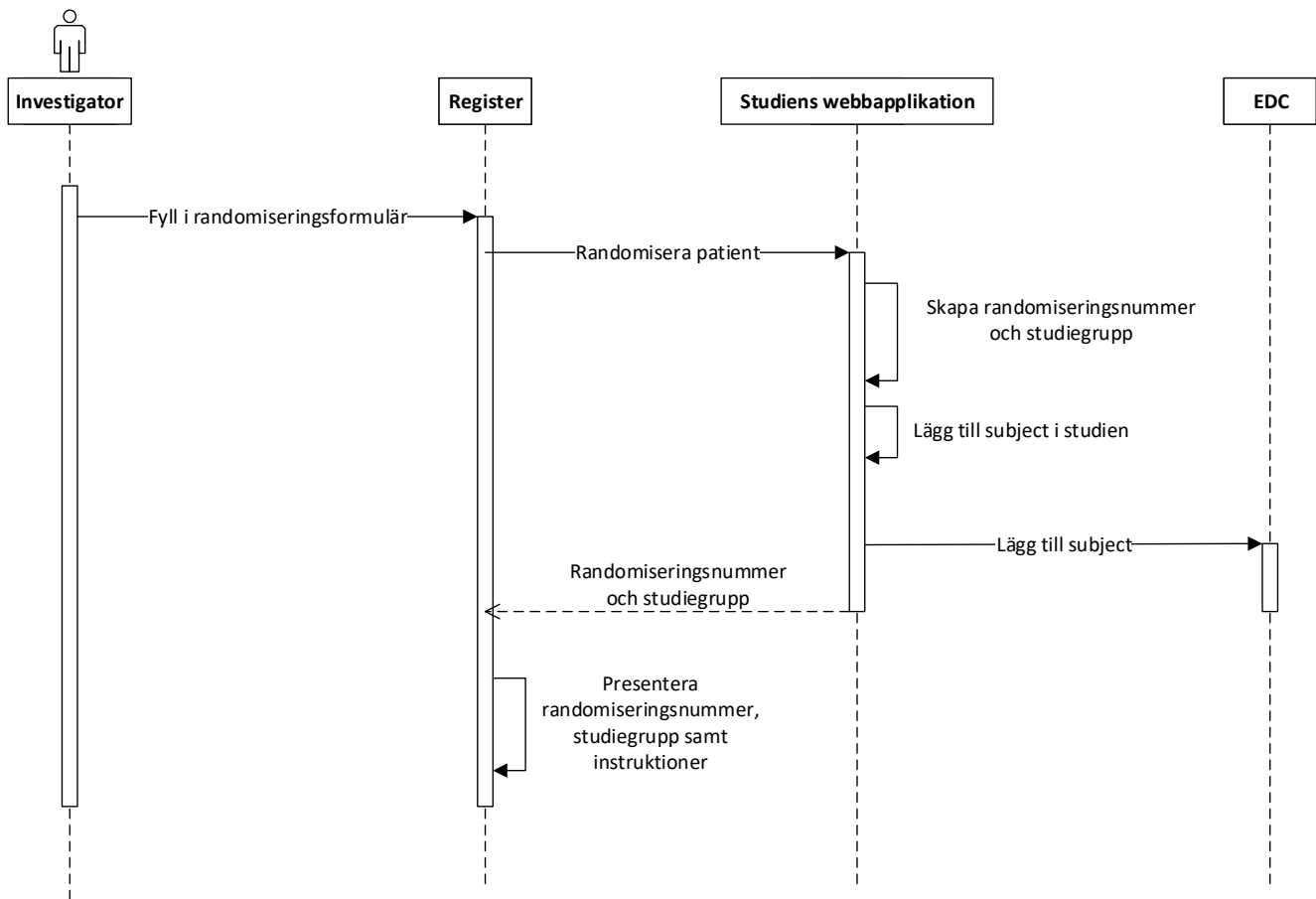
Figur 5. Komponenter i kvalitetsregistret. Studiespecifika komponenter i gult.



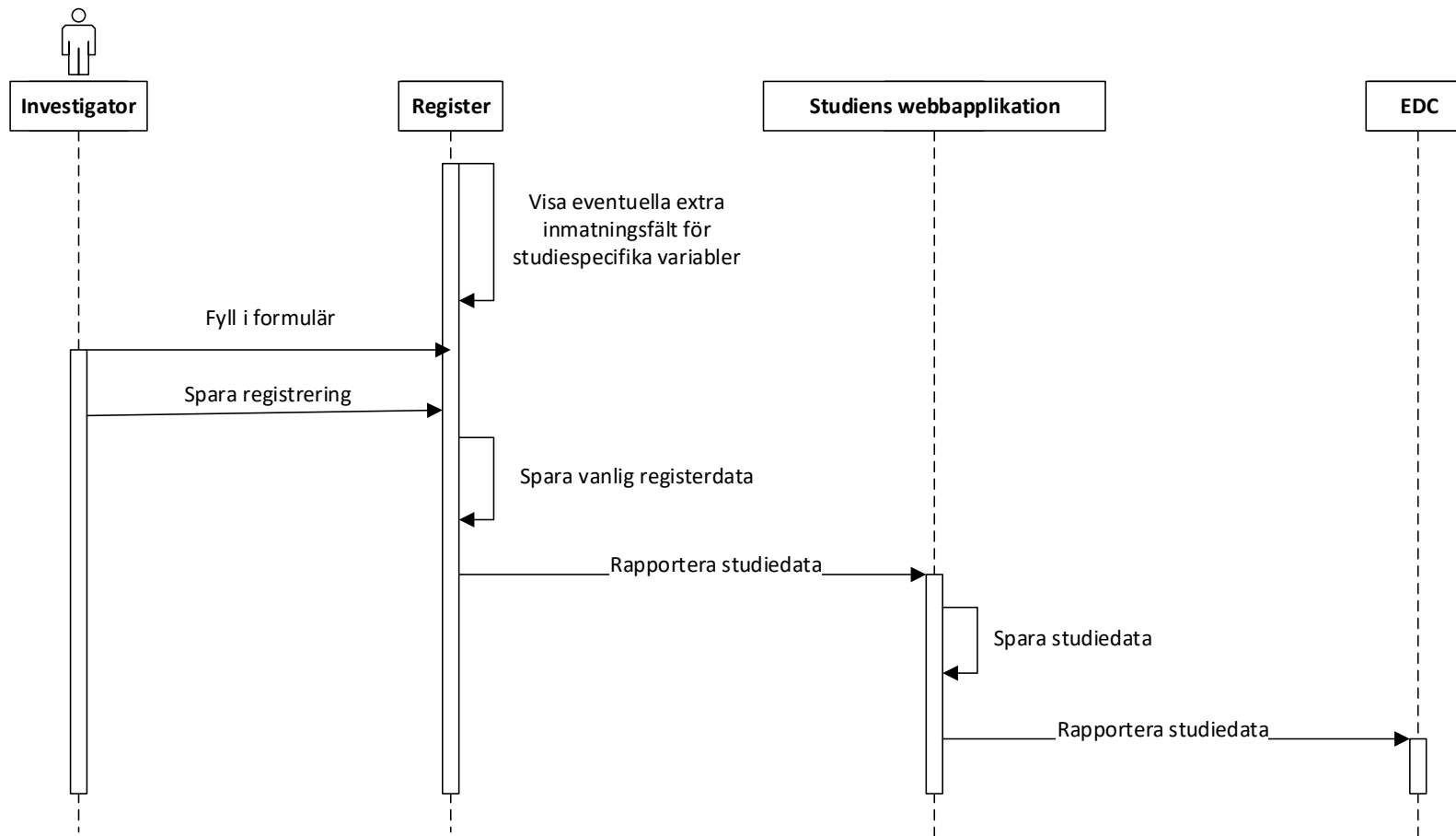
4.3.1. Randomisering och rapportering av studiedata

Följande två sekvensdiagram illustrerar randomisering och rapportering av studiedata från registret.

Figur 6. Sekvensdiagram för randomisering av patient.



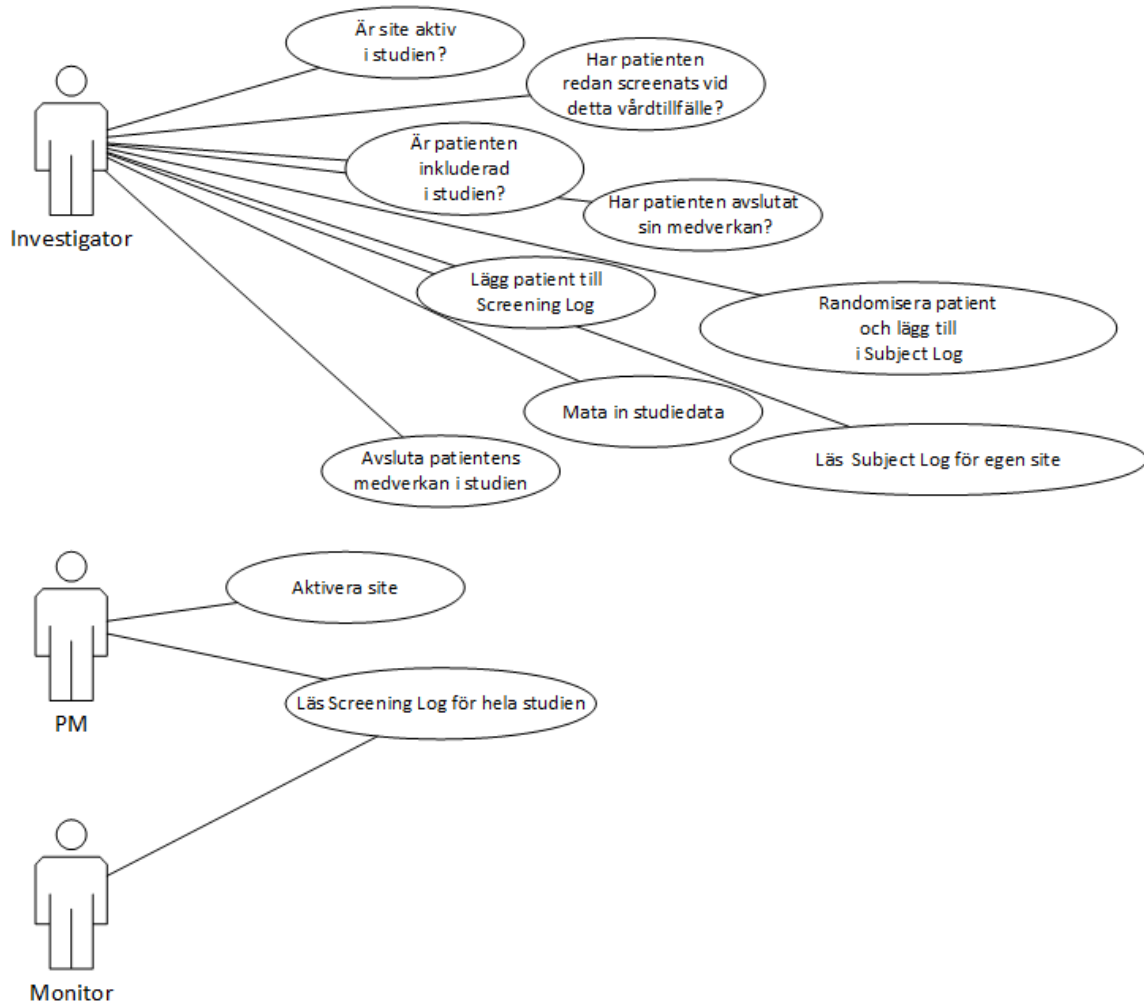
Figur 7. Sekvensdiagram för rapportering av studiedata.



4.4. Studiens webbapplikation

De tjänster som används av registret är vanligtvis screening, randomisering och rapportering av studiedata, men för att bedriva en studie behövs fler tjänster. Figur 8 illustrerar alla användningsfall som R-RCT-ramverket stödjer med REST-tjänster.

Figur 8. Användningsfall som stöds av R-RCT-ramverkets REST-tjänster.



5. Indataformat

Studiedata kan skickas in som en JSON-struktur i en http/POST. Varje enskild variabel adresseras med en id-sträng som representerar en utplattad form av den hierarki som definieras i studiens datamodell (i standardformatet CDISC ODM).

För att beskriva detta använder vi nedan ett exempel där en patient som deltar i en studie ska göra ett återbesök där man ska notera patientens vikt.

I studiens datamodell har man definierat händelsen SE_FOLLOWUP. Vid denna händelse ska provaren notera patientens vikt i variabeln I_WEIGHT. Variabeln I_WEIGHT återfinns i gruppen IG_FOLLO_UNGROUPED i version 1.2 av formuläret F_FOLLOWUP

5.1. Metadata i CDISC ODM-format

CDISC ODM är ett standardformat som används i kliniska studier för att beskriva data. Variabeln i exemplet definieras något förenklat så här enligt CDISC ODM:

```
<StudyEventDef OID="SE_FOLLOWUP">
    <FormRef OID="F_FOLLOWUP_12"/>
</StudyEventDef>

<FormDef OID="F_FOLLOWUP_12">
    <ItemGroupRef OID="IG_FOLLO_UNGROUPED"/>
</FormDef>

<ItemGroupDef OID="IG_FOLLO_UNGROUPED">
    <ItemRef OID="I_WEIGHT"/>
</ItemGroupDef>

<ItemDef OID="I_WEIGHT" DataType="integer" Length="3">
</ItemDef>
```

5.2. Variabelns id återspeglar hierarkin

CDISC-formatet är ganska utrymmeskrävande, så i R-RCT-ramverket plattar vi ut hierarkin ovan till en enkel sträng:

```
SE_FOLLOWUP::1::F_FOLLOWUP::1::IG_FOLLO_UNGROUPED::1::I_WEIGHT
```

Den här strängen kallas i R-RCT för variabelns id.

5.3. JSON-struktur i REST-anropet för lagring av data

Ett register som vill skicka in studiedata till R-RCT-ramverket använder det utplattade formatet ovan. Ett REST-anrop till R-RCT-ramverket för att lagra data kan därmed se ut ungefär såhär:

```
{
```

```

    "siteId" : "SE01",
    "sourceId" : "MittKvalitetsRegister",
    "reporterId": "DrSvensson",
    "items" : [
        {
            "id" :
"SE_FOLLOWUP::1::F_FOLLOWUP_12::1::IG_FOLLO_UNGROUPED::1::I_WEIGHT",
            "value" : "65"
        }
    ]
}

```

Här rapporteras alltså värdet 65 till variabeln WEIGHT i gruppen FOLLO_UNGROUPED i formuläret FOLLOWUP som används vid händelsen FOLLOWUP. Rapportörande provare har användar-id DrSvensson, och inmatningen gjordes i registret MittKvalitetsRegister på studiesitet SE01.

5.3.1. Ordningsnummer för repeterande komponenter

Händelser, formulär och grupper kan alla vara repeterande, det vill säga fler än en instans av samma typ av komponent kan skapas vid inmatning. När registret rapporterar studiedata är det registrets egna ordningsnummer som ska anges i variabel-id:t. I exemplet ovan har alla ordningsnummer värdet 1.

6. Arkitekturens mål och begränsningar

6.1. Modularitet

- En studie-webbapplikation ska kunna använda valda delar av R-RCT-ramverket enligt följande, utan att några av de artefakter som ingår i en R-RCT-release byggs om:
 - Endast hantering av subjects.
 - Hantering av subjects samt rapportering av studiedata till lokal databas.
 - Hantering av subjects samt rapportering av studiedata till EDC.
- Studiens datamodell ska kunna uppdateras utan att någon artefakt som ingår i ramverket byggs om.
- Integration mot ett EDC ska kunna byggas utan att någon artefakt som ingår i ramverket byggs om.

6.2. Testbarhet

Samtliga verksamhetsregler ska gå att testa på ett automatiserat sätt.

6.3. Begränsningar

- Ansvar för att sätta och uppnå mål för en R-RCT-implementations tillgänglighet ska ligga helt på det team som ansvarar för studien.
- Ramverket ska inte innehålla några delar som påverkar en färdig R-RCT-implementations prestanda på något avvikande sätt. Ansvar för att sätta och uppnå mål för prestanda ska ligga helt på teamet som ansvarar för studien.

- Ramverket ska inte innehålla några konstruktioner för skyddad åtkomst av vare sig tjänster eller lagrad information.
- Eftersom R-RCT-ramverket inte har något användargränssnitt direkt mot mänskliga användare finns inga mål för tillgänglighet uppsatta.
- Alla felmeddelanden och statusmeddelande som returneras av REST-tjänsterna är skrivna på engelska. Det går inte att utöka systemet med översättningar av dessa meddelanden till andra språk.

7. Kvalitet

I det här avsnittet beskrivs hur mjukvaruarkitekturen bidrar till att R-RCT-ramverket uppfyller de mål som satts upp i avsnitt 6.

7.1. Modularitet

Mål: En studie-webbapplikation ska kunna använda valda delar av R-RCT-ramverket enligt följande, utan att några av de artefakter som ingår i en R-RCT-release byggs om:

- Endast hantering av subjects.
- Hantering av subjects samt rapportering av studiedata till lokal databas.
- Hantering av subjects samt rapportering av studiedata till EDC.

Lösning: En modulär arkitektur och system för hantering av beroenden. Separata moduler hanterar de olika verksamhetsområdena och en R-RCT-implementation kan använda Spring Framework för att definiera vilka moduler som ska användas i den specifika implementationen (vilket praktiskt sett utgör konfigurationer).

Mål: Studiens datamodell ska kunna uppdateras utan att någon artefakt som ingår i ramverket byggs om.

Lösning: R-RCT-ramverket läser datamodellen från en fil i CDISC ODM-format och genererar nödvändiga representationer för att hantera objekten i systemet. Filen pekas ut med hjälp av en property som ska finnas tillgänglig på classpath i runtime. Datamodellen för studiens kliniska innehåll blir därmed en konfiguration som baserar sig på stödet för CDISC ODM

Mål: Integrering mot EDC ska kunna byggas utan att någon artefakt som ingår i ramverket byggs om.

Lösning: Samma som ovan. Även egenutvecklade kod för integrationer ska kunna hanteras genom system för att konfigurera använda moduler. Egenutvecklade delar kan därigenom bindas ihop med befintliga artefakter.