

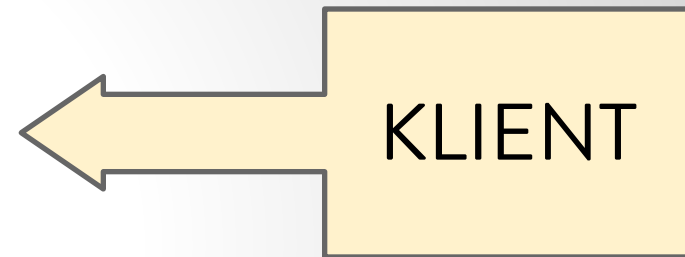
PAINTFEUD

Erfarenheter

OM OSS



Tobias Hill,
Citerus



Jesper Hammarbäck,
Citerus



AGENDA

- **VAD?**
Paintfeud, Spelet
- **VARFÖR?**
- **HUR?**
Arkitektur klient & server, CQRS
- **ERFARENHETER**

VAD?

PAINTFEUD

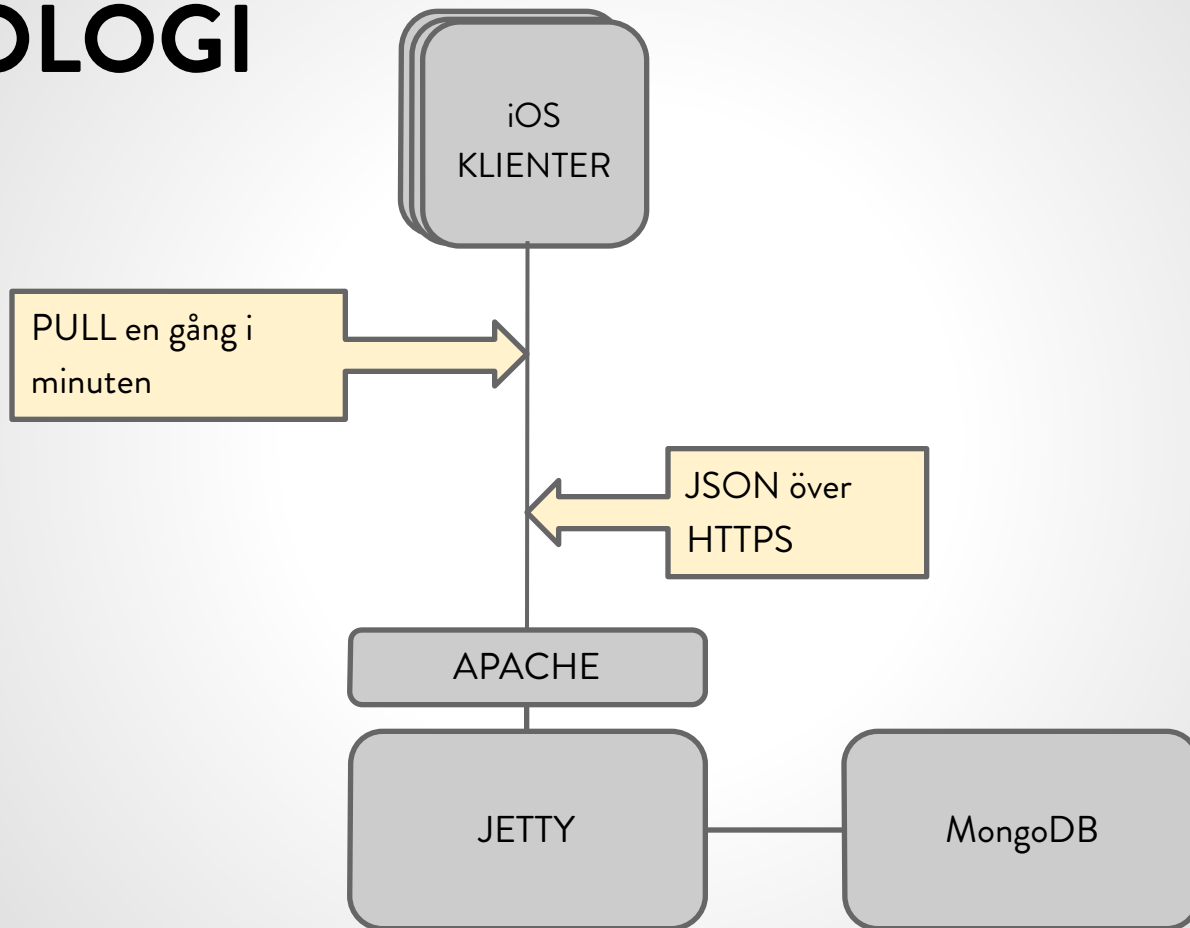
- Pictionary:ish
- 2-8 spelare per spel
- Spelarna turas om att rita och gissa
- Vinnare koras

VARFÖR?

- Utmaningen
- Lära & testa nya tekniker
- Så kul att vi inte kan låta bli

HUR?

TOPOLOGI



KLIENT

KLIENT

RAMVERK: Appcelerator Titanium

Javascript som *limspråk* mot rikt API

```
var view = Ti.UI.createView({
  backgroundColor: 'lightgray',
  layout: 'vertical'
});

var clicks = 0;

var field = Ti.UI.createTextField({
  borderStyle: Ti.UI.INPUT_BORDERSTYLE_ROUNDED,
  width: '30dp', height: '40dp', top: '20dp'
});

var button = Ti.UI.createButton({
  top: '20dp',
  title: 'Click me!'
});

button.addEventListener('click', function () {
  field.value = ++clicks;
});

view.add(field);
view.add(button);

var window = Ti.UI.createWindow();
window.add(view);
window.open();
```



KLIENT

RAMVERK: Appcelerator Titanium Pluginbaserat

```
var QG2d = require('com.googlecode.quicktgame2d');

var gameView = QG2d.createGameView();
var scene = QG2d.createScene();

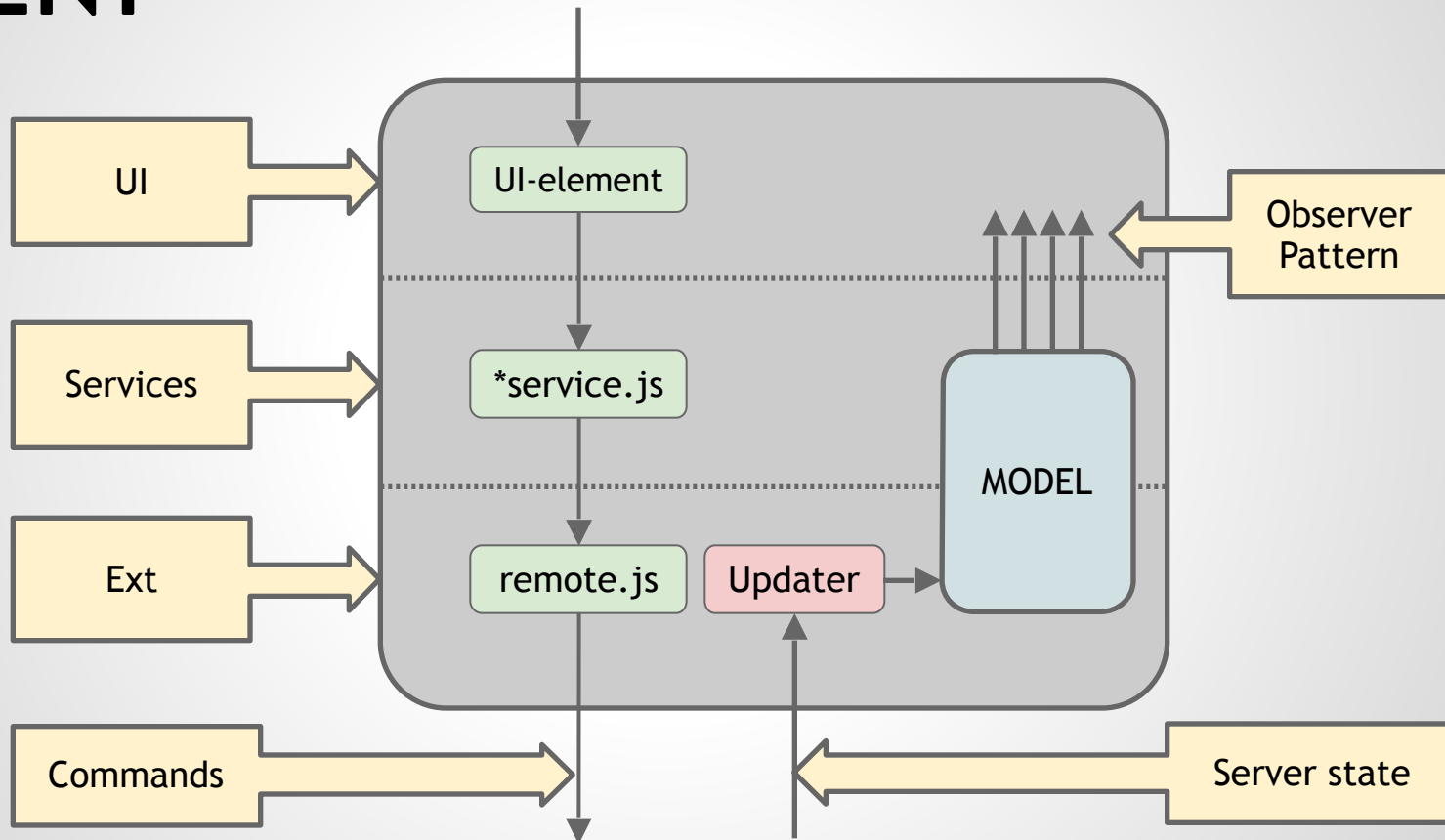
gameView.pushScene(scene);

gameView.addEventListener('onload', function () {
    gameView.start();
    var emitter = QG2d.createParticles({
        image: 'graphics/forestfire.pex'
    });
    emitter.move({
        gameView.screen.width * 0.5,
        gameView.screen.height * 0.5
    });
    scene.add(emitter);
});

var window = Ti.UI.createWindow();
window.add(gameView);
window.open({fullscreen:true, navBarHidden:true});
```



KLIENT



SERVER

CQRS

Command Query Responsibility Segregation

- B. Meyers CQS, Command/Query Separation
Att ställa en fråga ska aldrig ändra svaret
- Tillämpas inom ett Bounded Context

CQRS

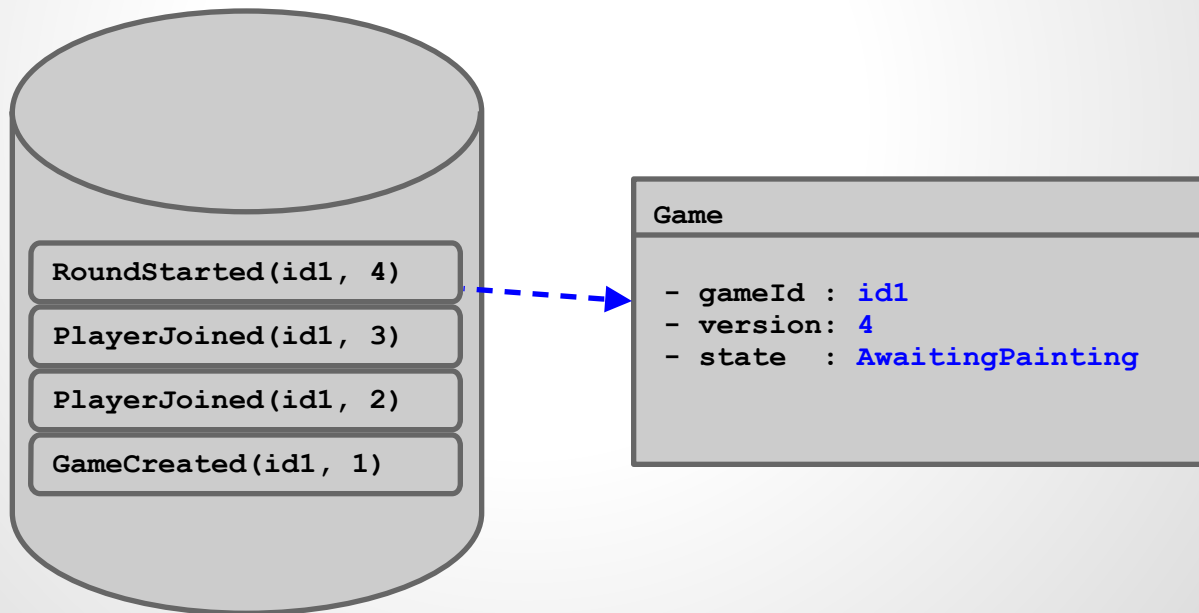
"Simply the creation of two objects where there was previously only one"

-- Greg Young

- Separerar skriv från läs inom kontextet
Ren domänmodell på skrivsidan
Flera klientanpassade modeller på lässidan

EVENT SOURCING

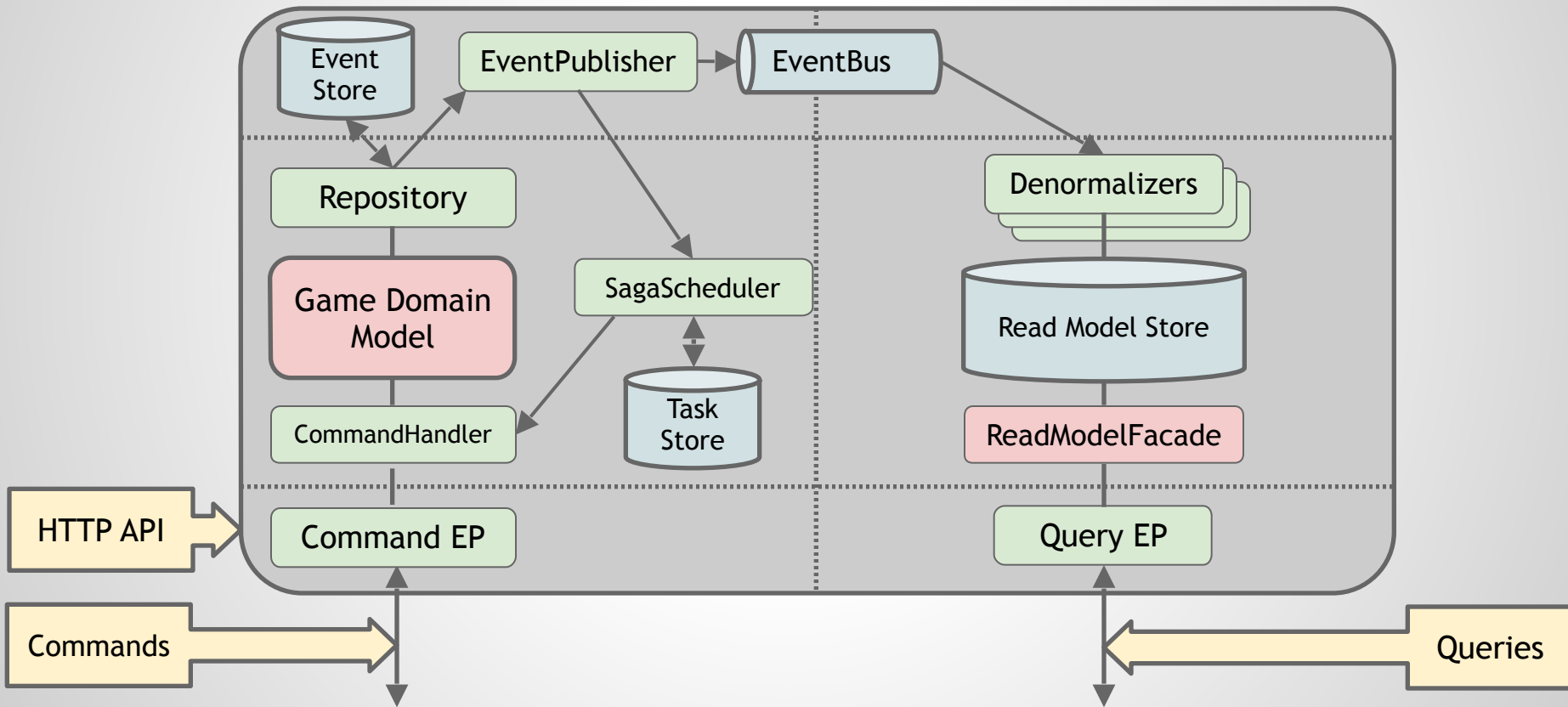
Definition: *Capture all changes to an application state as a sequence of events*



SERVER

- Scala
- Scalatra
- Spring (DI, i18n, JMX etc.)
- Jetty
- Quartz Scheduler
- Maven

SERVER - GAME BC



VARFÖR CQRS+ES?

ERFARENHETER

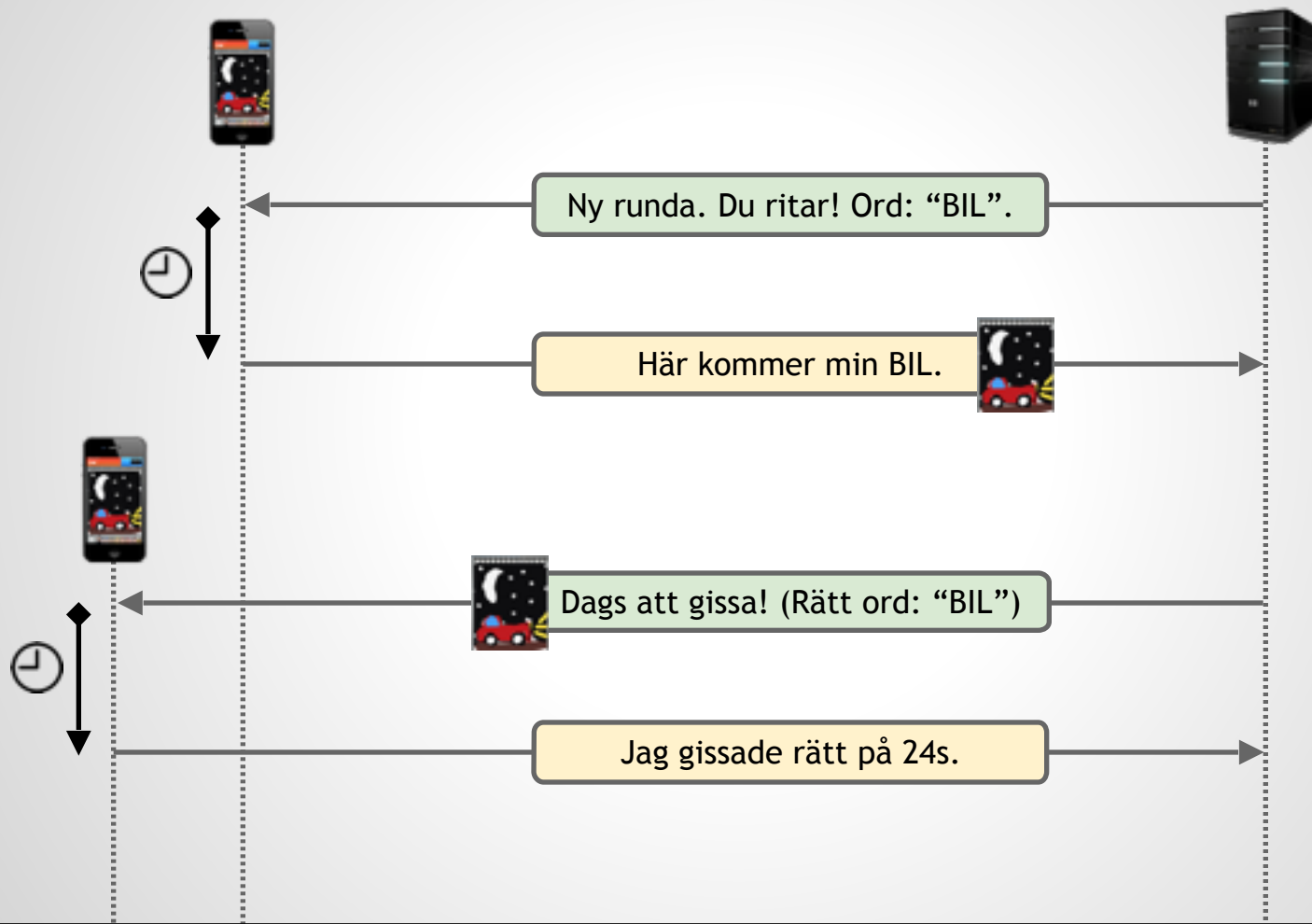
ERFARENHET 1:
**BRA KLIENTUPPLEVELSE
ÄR SVÅRT**



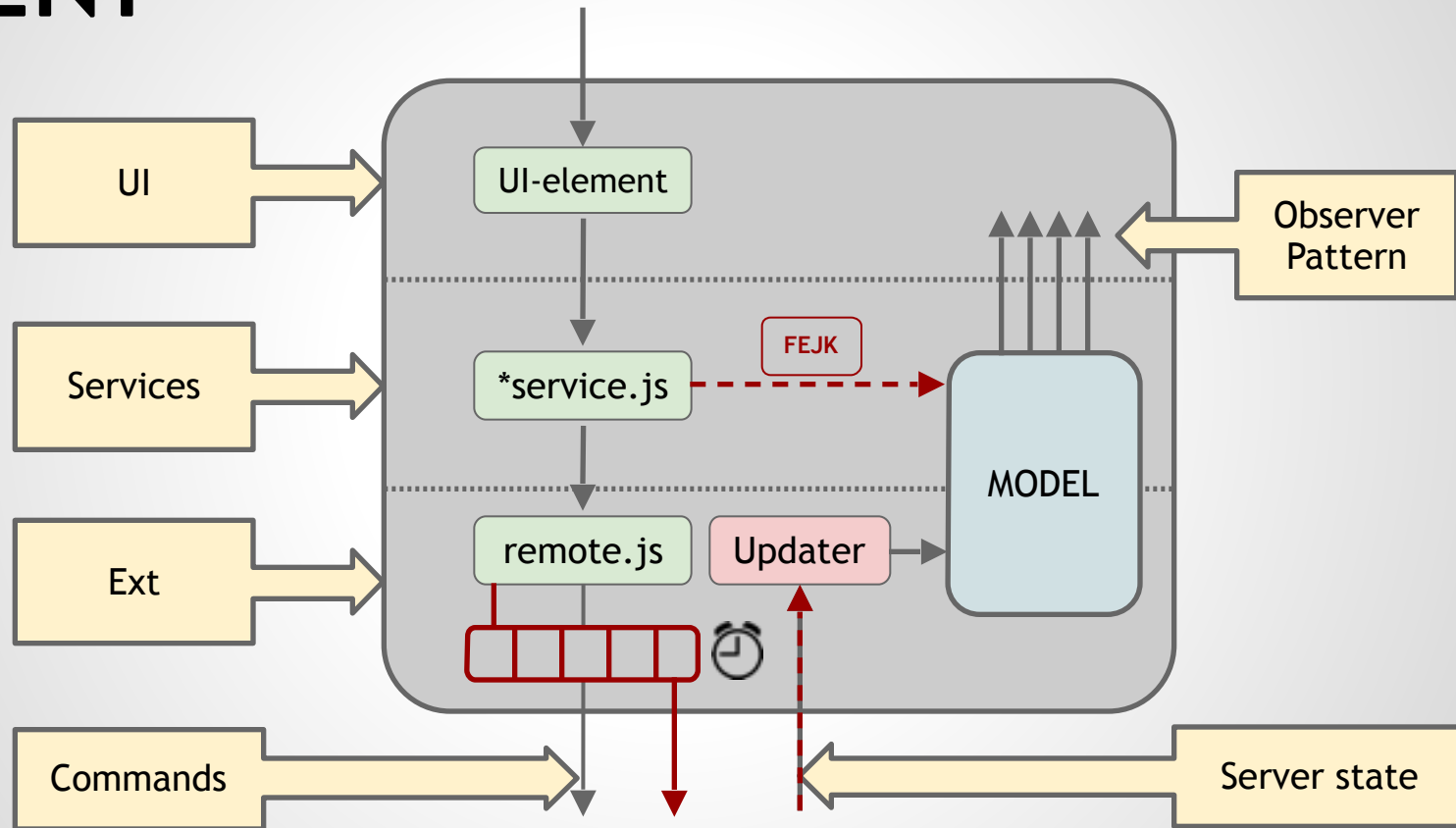
~~ANTAGANDE:~~

~~“Mobilnätverk är tillräckligt bra”~~





KLIENT



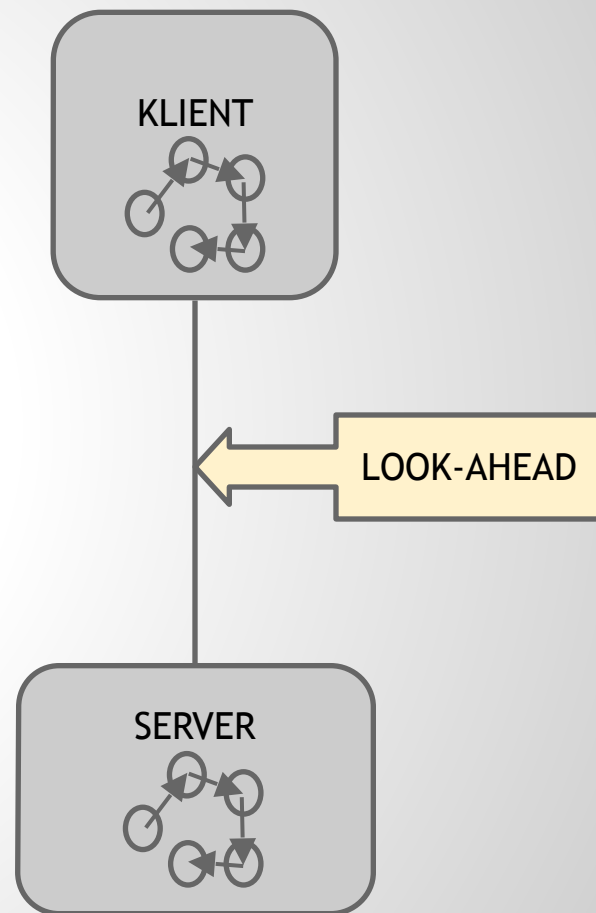
STATE

EXEMPEL PÅ FEJK (för flow)

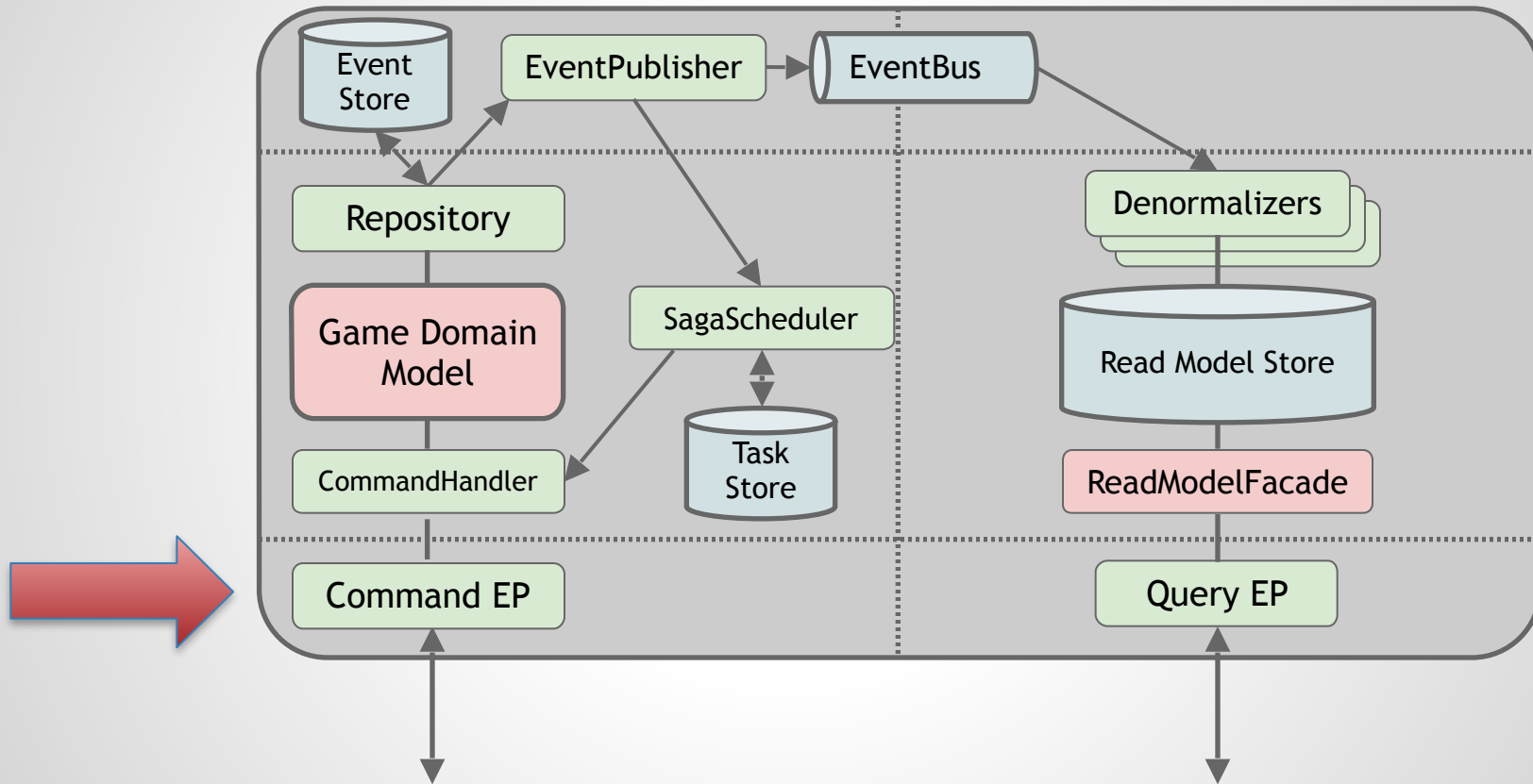
- Spelare accepterar/nekas ett spel.
- Spelare har just ritat eller gissat.

STÖRRE UTMANING

- Spelare är sist att acceptera/neka ett spel.
- Spelare X gissar sist i en runda, men ritas inte i nästa.
- Spelare X gissar sist i en runda, och är den som skall rita i nästa.

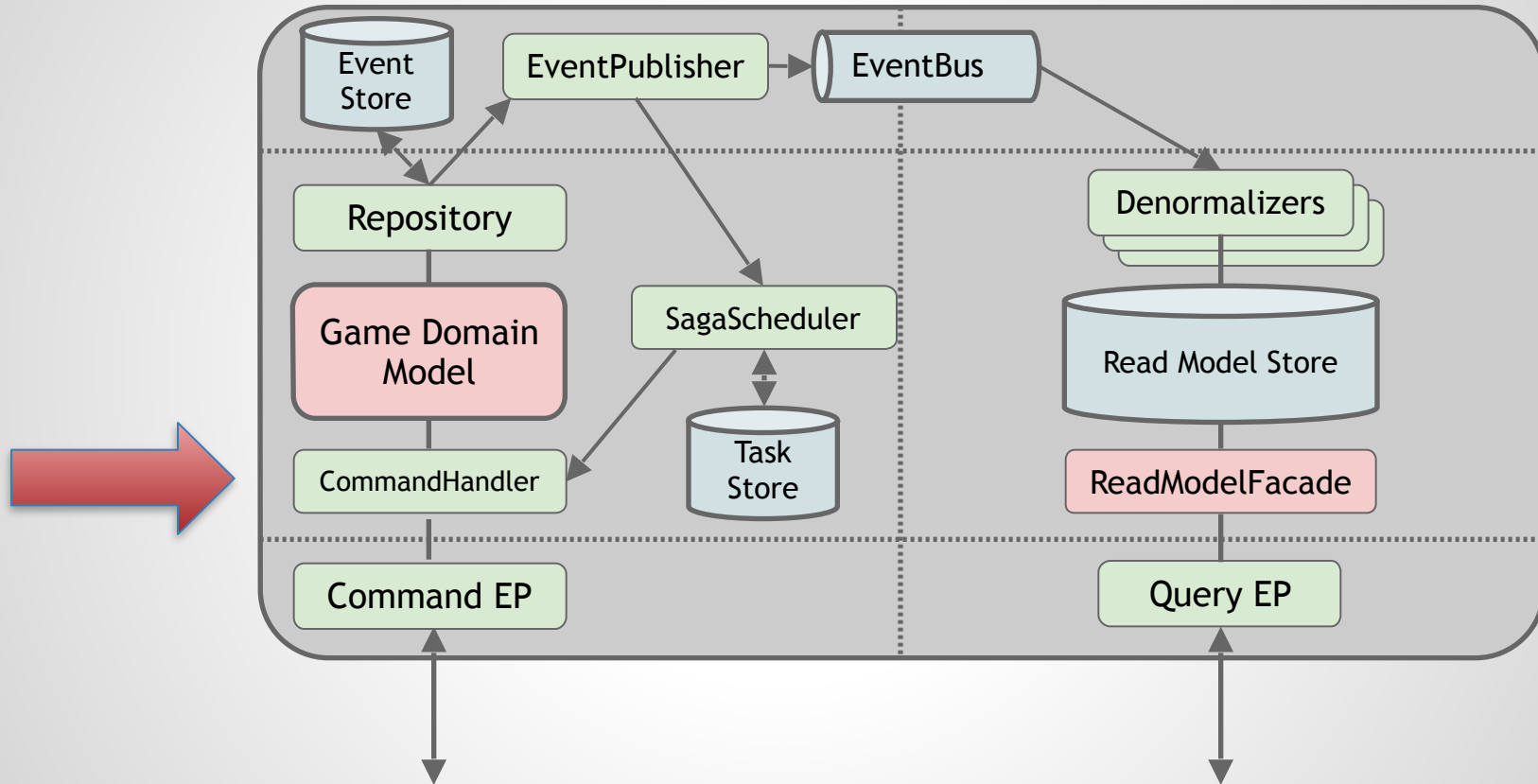


ERFARENHET 2: **CQRS GER FÖRDELAR**



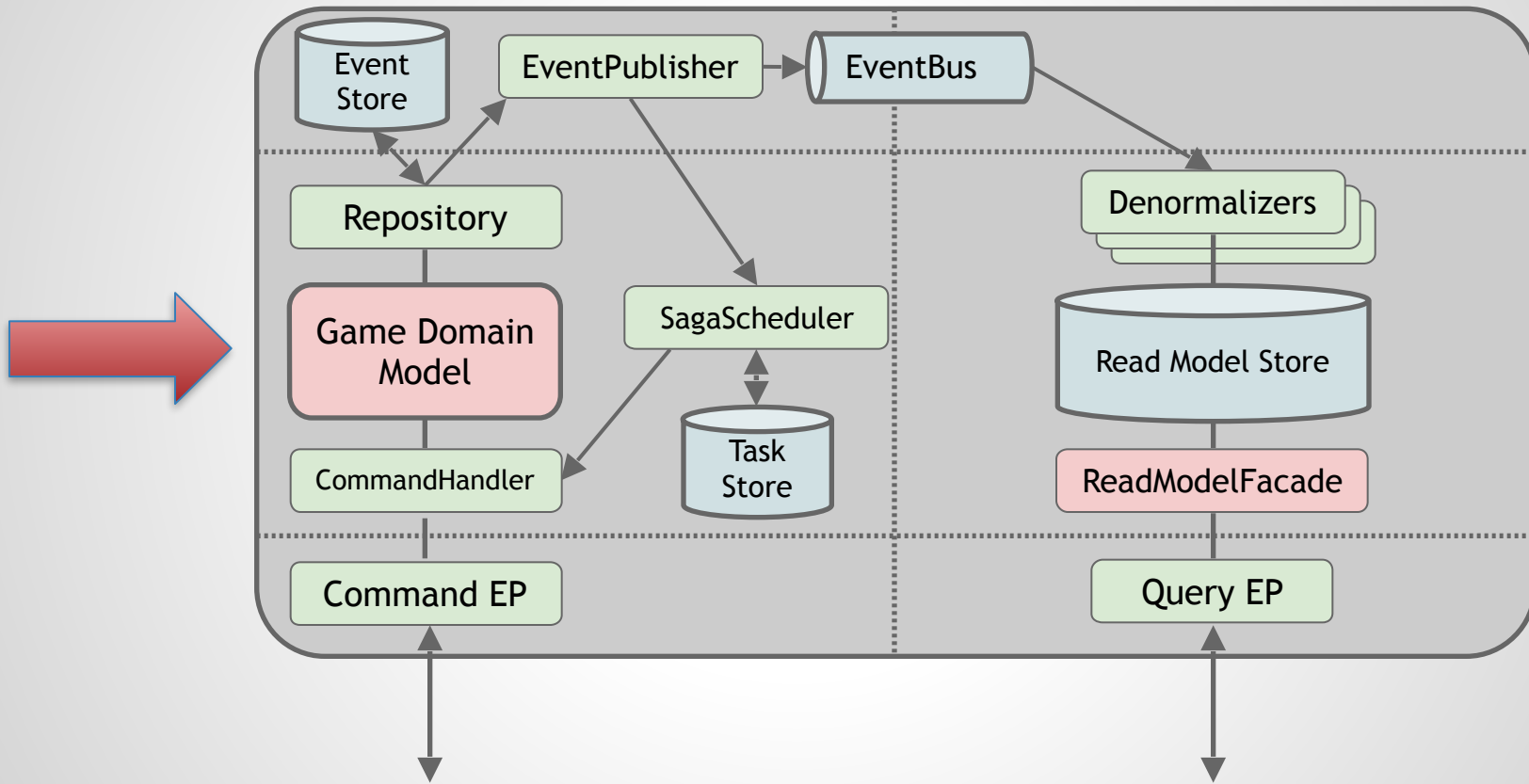
HTTP-API

```
post("/:gameId/join") {  
  
    // May cause IllegalArgumentException -> SC_BAD_REQUEST (400)  
    val gameId = gameId(params("gameId"))  
    val sessionId = sessionId(params("sessionId"))  
  
    // May cause NoSessionException -> SC_UNAUTHORIZED (401)  
    val playerId = sessionRepository.get(sessionId)  
  
    commandHandler ! JoinPlayer(gameId, playerId)  
    status(SC_OK)  
}
```



CommandHandler (Actor)

```
case JoinPlayer(gameId, playerId) => {  
  val game = repository.load(gameId)  
  game.joinPlayer(playerId)  
  repository.save(game)  
}
```

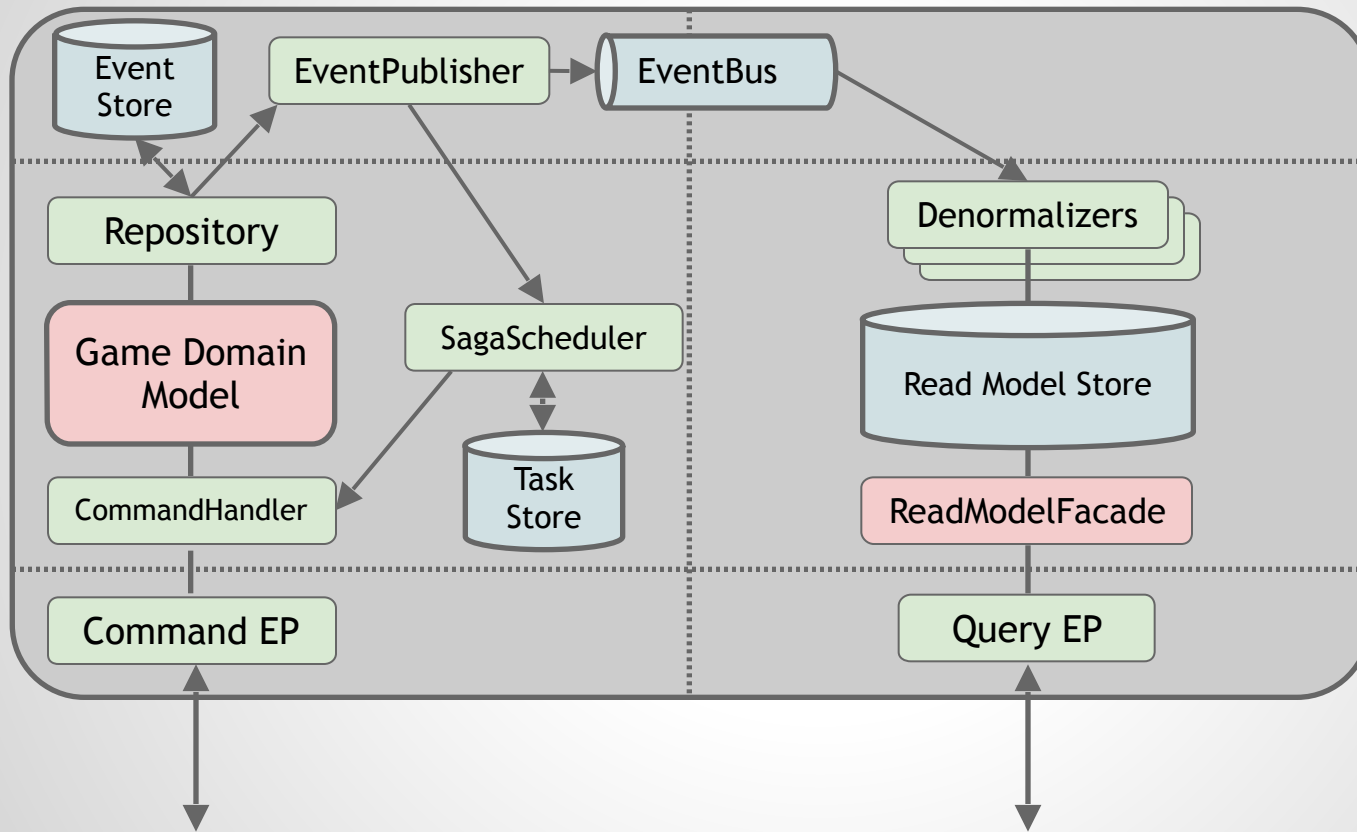
GAME - AGGREGATROT

```
class Game extends AggregateRoot {  
  
  private val joinedPlayers = new LinkedHashSet[PlayerId]  
  
  def joinPlayer(playerId: PlayerId) {  
    assertIsInvited(playerId)  
    if (!hasAlreadyJoined(playerId)) {  
      applyChange(PlayerJoined(gameId, nextVersion(), now(), playerId))  
      if (hasReceivedAllInvitationResponses) {  
        applyChange(RoundStarted(gameId, nextVersion(), now(), creator, nextWord()))  
      }  
    }  
  }  
  
  def handleEvent(event: PlayerJoined) {  
    version = event.version  
    lastUpdated = event.timestamp  
    joinedPlayers += event.playerId  
  }  
}
```

GAME - AR TRAIT

```
trait AggregateRoot {  
  
    private val uncommittedEvents = new ListBuffer[GameEvent]  
  
    def applyChange(e: GameEvent, isNew : Boolean = true) {  
        getClass.getMethod("handleEvent", e.getClass).invoke(this, e)  
        if (isNew) uncommittedEvents += event  
    }  
  
    def loadFromHistory(history: List[GameEvent]) {  
        history.foreach(applyChange(_, isNew = false))  
    }  
}
```

**EN REN MODELL ÄR
LÄTT ATT TESTA**

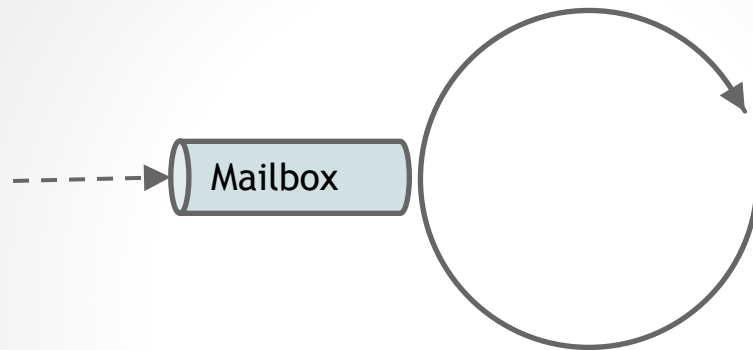


QUERY-SIDAN - DENORMALIZERS

- Enkelt att lägga till nya vid behov
- Undvik beroenden mellan denormalizers
- Bygg upp nya modeller mha återpublicering

ERFARENHET 3: **ACTORS FÖRENKLAR**

ACTORS



```
def getHandler(id: GameId) = {  
  handlerPool(math.abs(id.hashCode % numberOfHandlers))  
}
```


ERFARENHET 4: **DYNAMISKT DATA**

- Lägg till dynamisk data vid query-tillfället
- *Avatar, antal chattmeddelanden etc.*

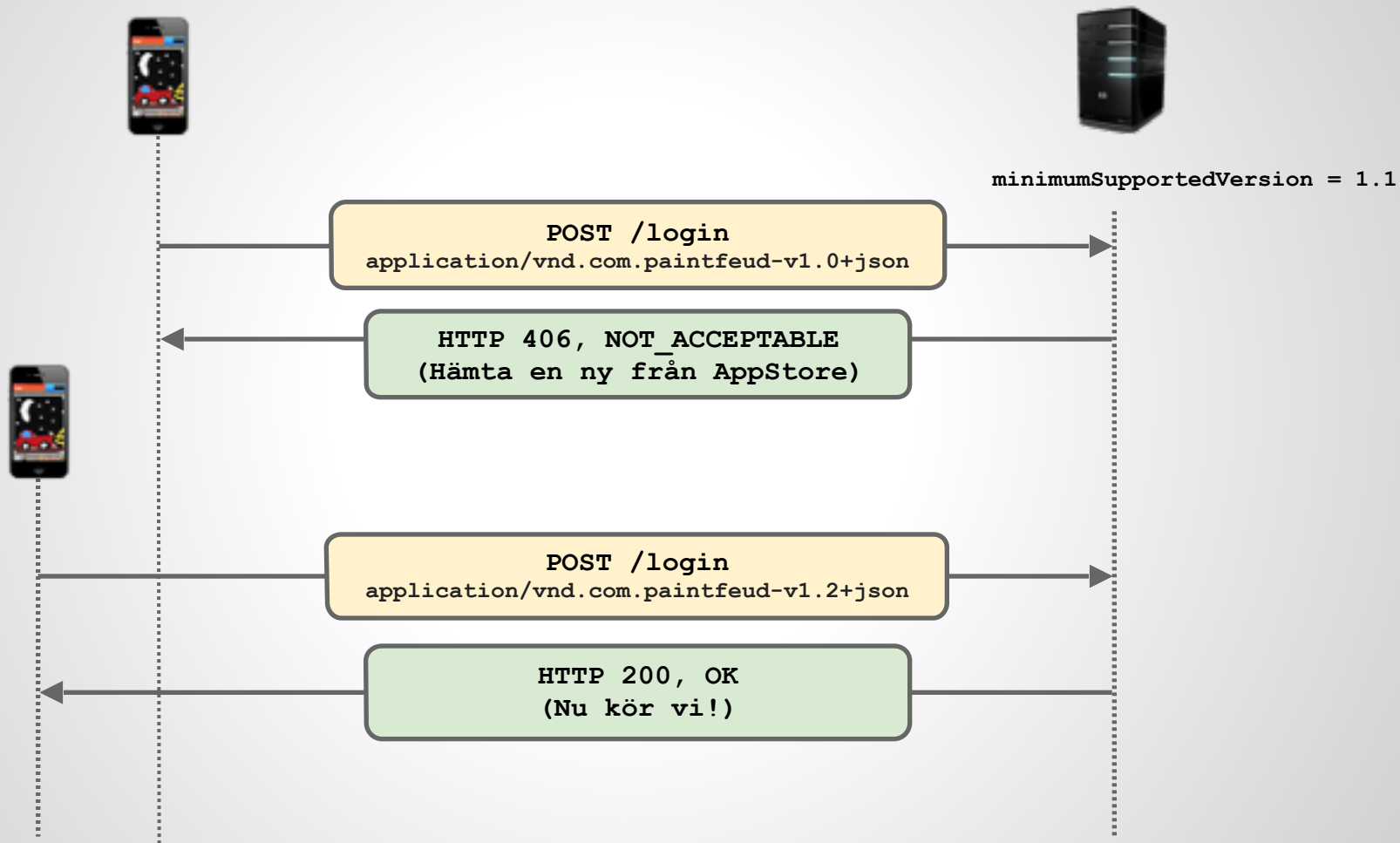
ERFARENHET 5: **EVOLVERANDE LOGIK**

FÖRÄNDRING AV EVENTS

- Scalas Option (JSON-deserialiseraren använder *None*)
- Ny konstruktor (JSON-deserialiseraren väljer den som passar)
- Nytt event, t.ex ***PlayerJoinedWithLocale***

ERFARENHET 6: **VERSIONSHANTERAT API**

- Undvik om möjligt, pga ökad komplexitet
- JSON är naturligt förändringståligt
- Vi versionshanterar mha HTTP Accept header



SAMMANFATTNING

Tobias Hill

tobias.hill@citerus.se

Spelarnamn i Paintfeud: tobias

Jesper Hammarbäck

jesper.hammarback@citerus.se

Spelarnamn i Paintfeud: jesper