



AARHUS UNIVERSITET

Software Architecture in Practice

Architectural Prototyping Exercise

Henrik Bærbak Christensen



- We have a “small scale” MMORPG on Web
 - “WoW light”
- Business model
 - Play via browser
 - Small part of world free / restricted artifacts
 - Subscription/fee for whole world / all artifacts
- Problem
 - Success
 - Too many players, only one game server
 - Gamers often experience lag, because server is overloaded

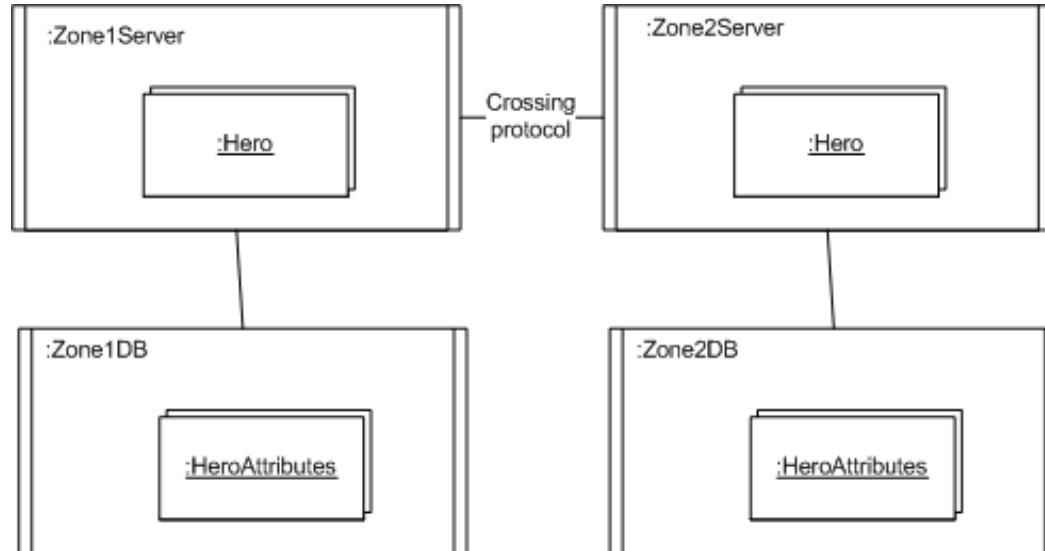
The challenge



- **Tactic: Introduce concurrency ?**
 - Multiple servers
 - Divide virtual world into 16 geographical zones; one zone = one server + associated database server for that zone
 - Risk: how fast can we transfer a (lot of) character(s) from one server to another, when a group of players cross a zone boundary
 - Or put in another way: will it influence game experience?
- **QAS**
 - *1000 avatars cross world zone A to B within 1 second under normal conditions, all players experience a max “freezing” of movement of 1 second / all data is transferred + persisted within 1 second*

A Hypothetical CC view

- What does the Crossing Protocol entail?
 - Hero object moves from one server to another
 - Hero attributes/data moves from one database to another
 - And client redirected to another server (hypothesis: via an URL)



Or Tentative Code

- `hero.move(x1,y1)` (state is `(x0,y0)`)
 - `(x0, y0)` belongs to Zone A
 - `(x1, y1)` belongs to Zone B
 - ... then transfer hero to Zone B server from Zone A server
- Algorithm hypothesis:

```
– public URL move(int x1, int y1) {  
    if (isCrossingZones(x0, y0, x1, y1)) {  
        return transferToZone(x1, y1);  
    }  
    // normal movement algorithm  
    return null; // indicate no zone change  
}
```

Note: Client API is changed, but that is of no concern here...



Process A

AARHUS UNIVERSITET

- We have heated arguments on the white board and the ‘world zone’ priets win the battle (those who think it will not work get tired of the argument...)
- Thus we implement it directly in our game code
 - Introduce extra servers with associated game DBs
 - Rewrite large portions of the avatar movement code
 - *What if it turns out as a really bad idea?*



Process A2

AARHUS UNIVERSITET

- *You think that world zoning architecture is a lot of b.....*
 - *It is too cumbersome to program and maintain*
 - *It will not be fast enough for fluent play*
 - *It may generate random load peaks on servers*
- *And you want to **prove** that you are right*



Proposal B

AARHUS UNIVERSITET

- Prototype an *architectural exploratory prototype*
 - Or rather – harvest one from the existing system...
- Task:
 - Plan an AP to validate this tactics feasibility
 - What data from the real system do you need to measure?
 - What will a minimal code base have to include to experiment with the proposed architecture?
 - How to measure and validate hypothesis in the AP?
 - What about the work load of the servers?
 - What technology to use?



Exercise Premise

- The design is a dictate by me
 - Other designs are probably better, but this is not the point!
- You job
 - Faithfully create AP for this design
 - And if you dislike it, happily demonstrate that it is a very bad design 😊

- Prototype an architectural exploratory prototype
- Task:
 - Plan an AP to validate this tactic's feasibility
 - What data from the real system do you need to measure?
 - What will a minimal code base have to include to experiment with the proposed architecture?
 - How to measure and validate hypothesis in the AP?
 - What about the work load of the servers?
 - What technology to use?

