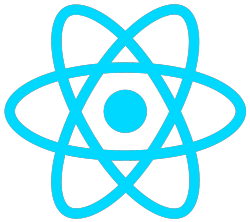
Securing the monolith

Marko Jurisic
@JurisicMarko

JavaCro'21

Intro



Motivation

- 20 year old monolith
- self made security
- planned new features (wallet, p2p payments)
- standard solution - RedHat SSO

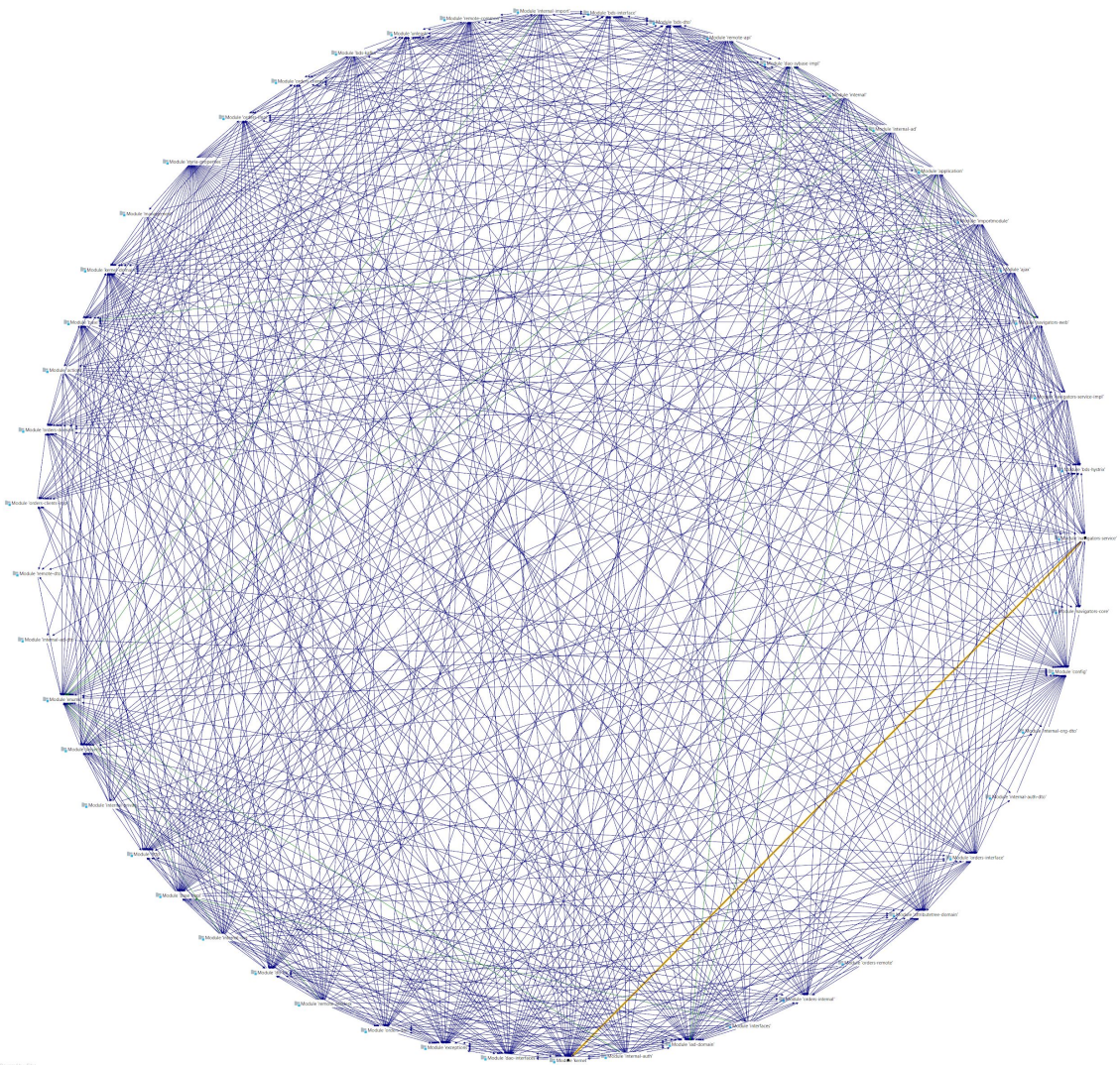
Constraints

- fixed go-live date
- seamless user migration (6.5 million users)
- all logged in users stay logged in
- gradual rollout (unleash)
- rollback scenarios
- everything stays better

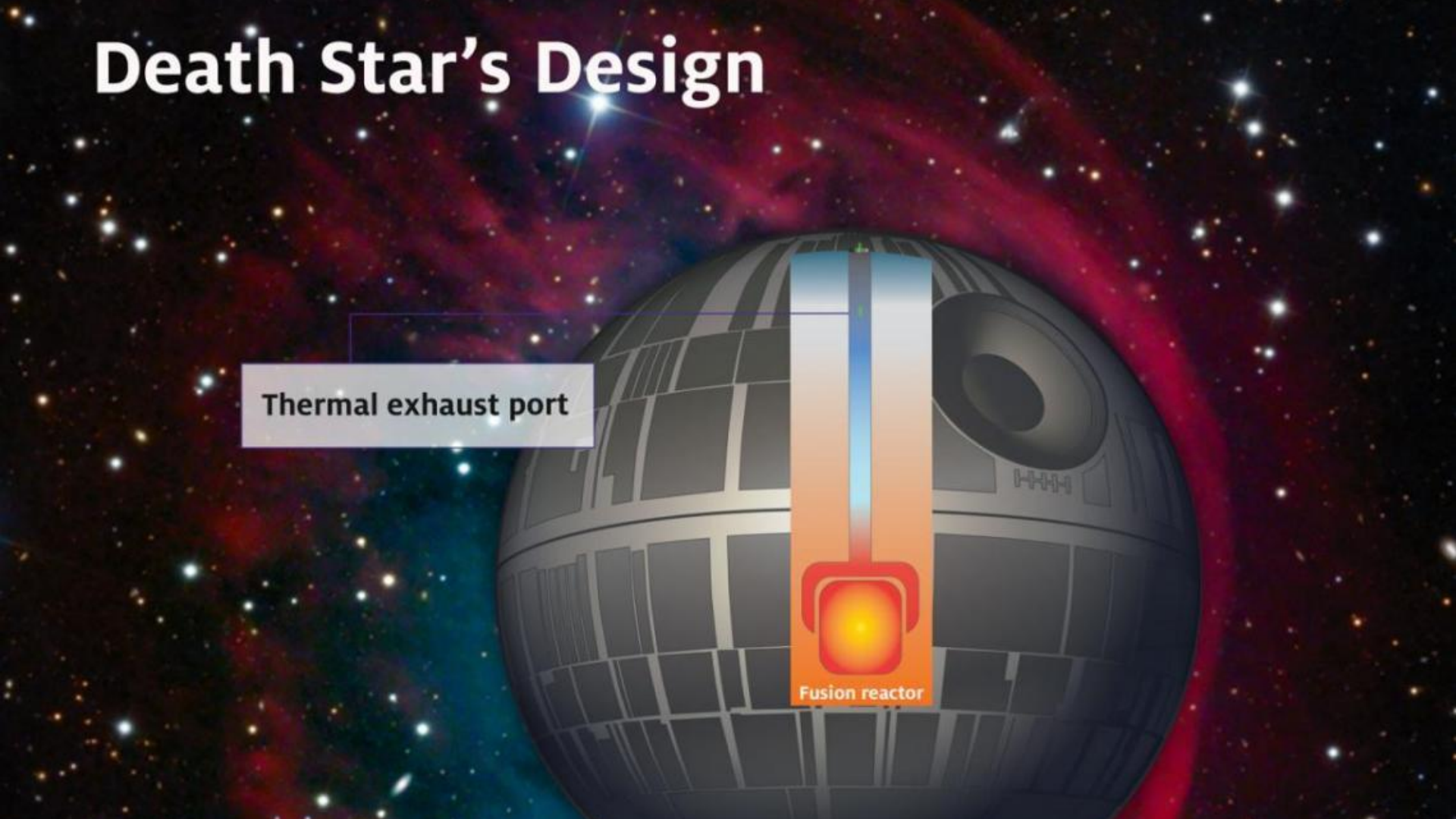
No-Goes

- big bang release
- user logout

Modules



Death Star's Design



Project plan

- start with business users (15k)
- a month or two
- do full rollout for web in about three months
- apps immediately after desktop

Integration - first try

- business users login (15k)
- setup realm, clients and roles
- test migration and synchronization
- first hurdles with non-standard behaviors
- extension with SPIs (Service Provider Interface)

Challenges

- platform size
- 2 way real time synchronization
- REST API => kafka
- impossible to completely separate user groups

Business user go-live in November 2019 🎉

User migration

Plan

- re-hash users passwords from production database
- import users into Redhat SSO
- perform tests
- ~~party~~

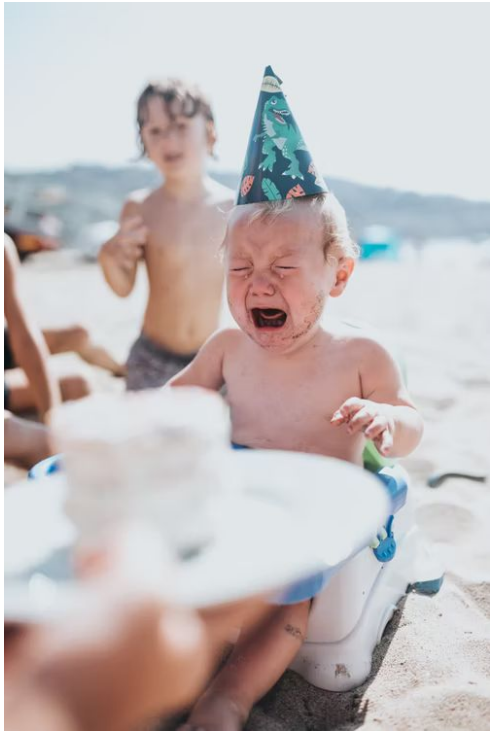


Photo by Nathan Dumlao on Unsplash (source)

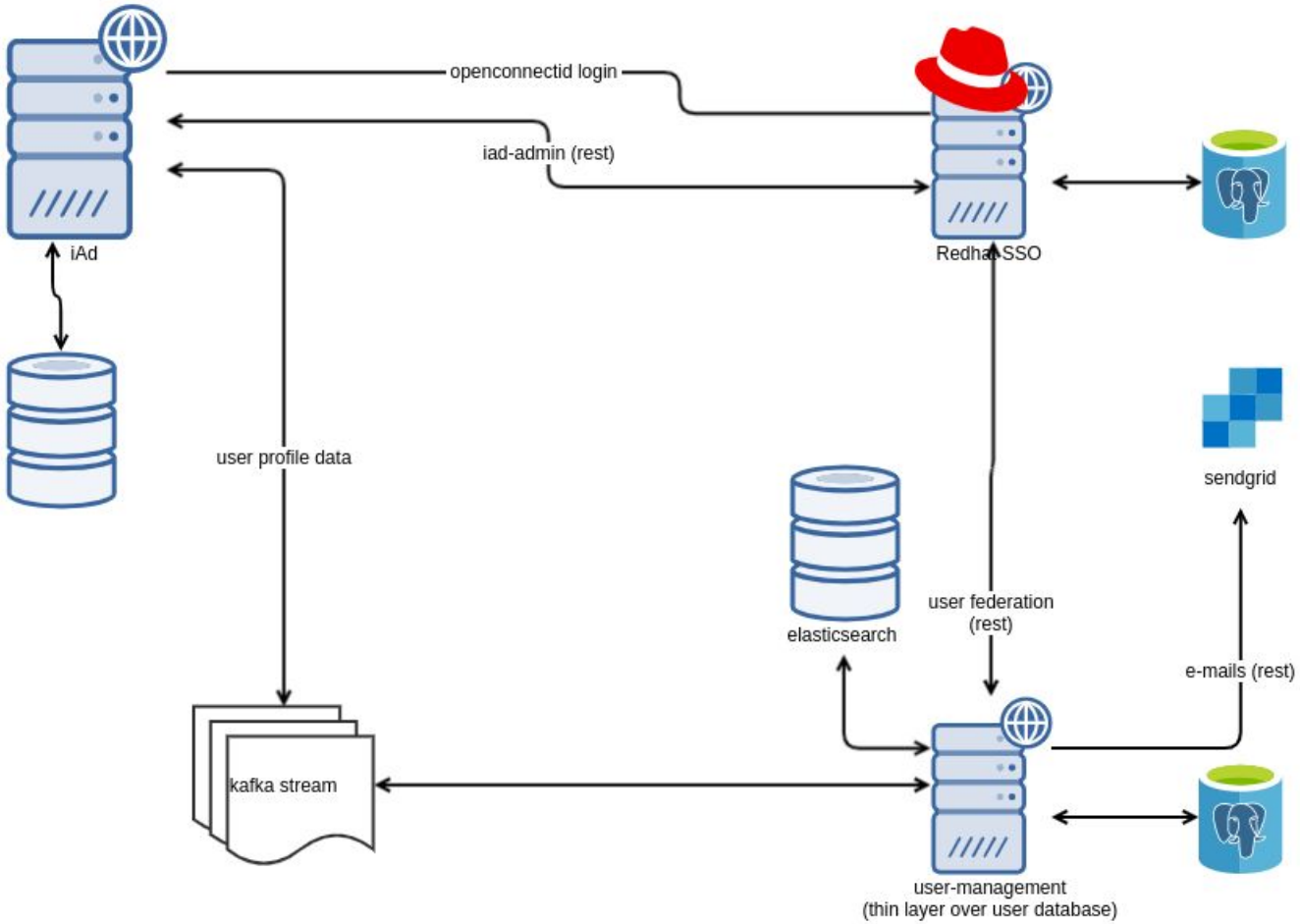
Integration - second try

User federation

- keycloak's way of fetching user data from an external source
- commercial solutions?
- write our own component!

User management

- lightweight component written in Kotlin
- Spring Boot and Spring Data
- just a thin layer providing REST interface to the new database


openconnectid login
iad-admin (rest)
iAd
RedhatSSO
user profile data
sendgrid
elasticsearch
user federation
(rest)
e-mails (rest)
kafka stream
user-management
(thin layer over user database)

Release

Testing

- functional tests
- security audits (2x)
- performance tests (login, token refresh, db)
- feature toggle tests (+rollback)
- integrity checker
- release!

Release

- gradual rollout using unleash feature toggles
- no explosions

Mobile rollout

App release

- 80% of our traffic comes from app users
- use android staged release instead of feature toggles
- routine, no need for a rollback scenario
- still had to support both auth methods on the backend because of legacy apps
- release went fine
- party time!

Photo by Afif Kusuma on Unsplash (source)

Mobile rollout

Showstopper

- small change, simple release
- **everything** died
- no panic, no finger pointing
- service restored after 8 hours (backups ftw!)
- restore service and analyze causes

Photo by Artur Kornakov on Unsplash (source)

Post mortem

- better resilience, hystrix circuit breakers
- root cause analysis
- redhat SSO cold start
- redhat support ping-pong
- infinispan

Final state

- custom session management
- completely stateless
- switch from Redhat SSO to keycloak
- two keycloak clusters in dislocated data centers
- upgrades without downtime
- it works the same, just better

Photo by Benjamin Davies on Unsplash (source)

WILLHABEN

Thank you!
Good Luck!