

You've got Microservices Now Secure Them!

Steve Pousty (@TheSteve0)

Stian Thorgersen (@stian_st)

Red Hat - we do fun things

<http://bit.ly/j1ddslides>

Goals

- Talk a little about Microservices
- A little about centralized authentication
- See it in Action

Microservices



PetitPlat.fr - sk.



ARCHDUKE
CHARLES

Hohenzollern

Rosenberg

Nordmann

Markgrainneusiedel

Russbach

WAGRAM

Bellegarde

Baumersdorf

Bernadotte
Saxons

Oudinot

Davout

Adersdorf

Kollowrath

Sussenbrunn

Massena

Raasdorf

NAPOLEON

Lauriston

Guard

Grenier

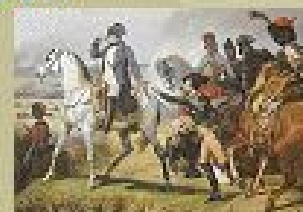
Marmont

Bessieres

MacDonald

Breitenlee

Klenau



Wrede
Bavarians

Essling

Aspern

Heavy
artillery

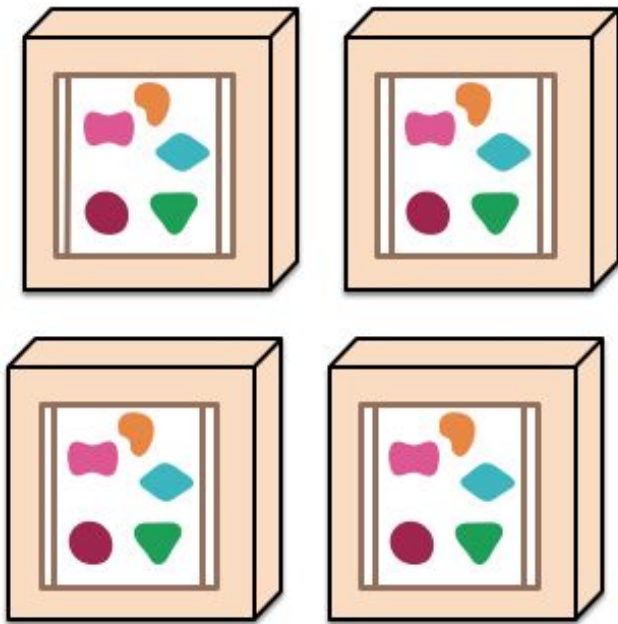
Bridges

Copyright 2007
Napoleon, His Army -
And Enemies.

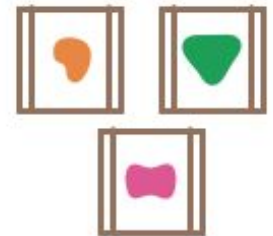
A monolithic application puts all its functionality into a single process...



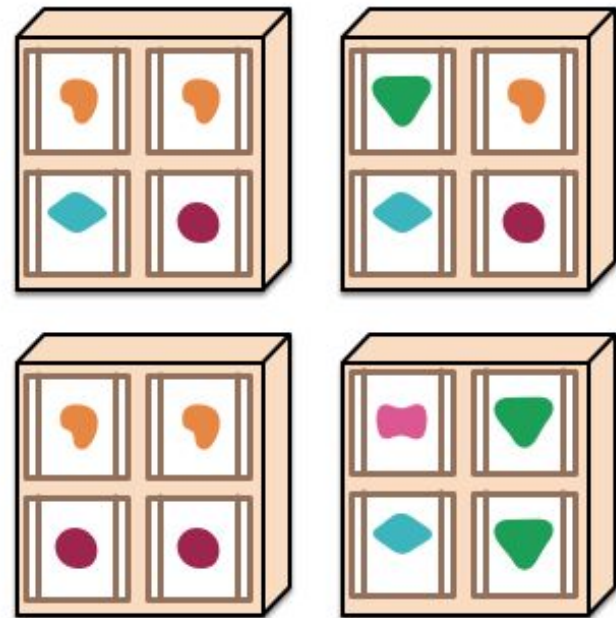
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



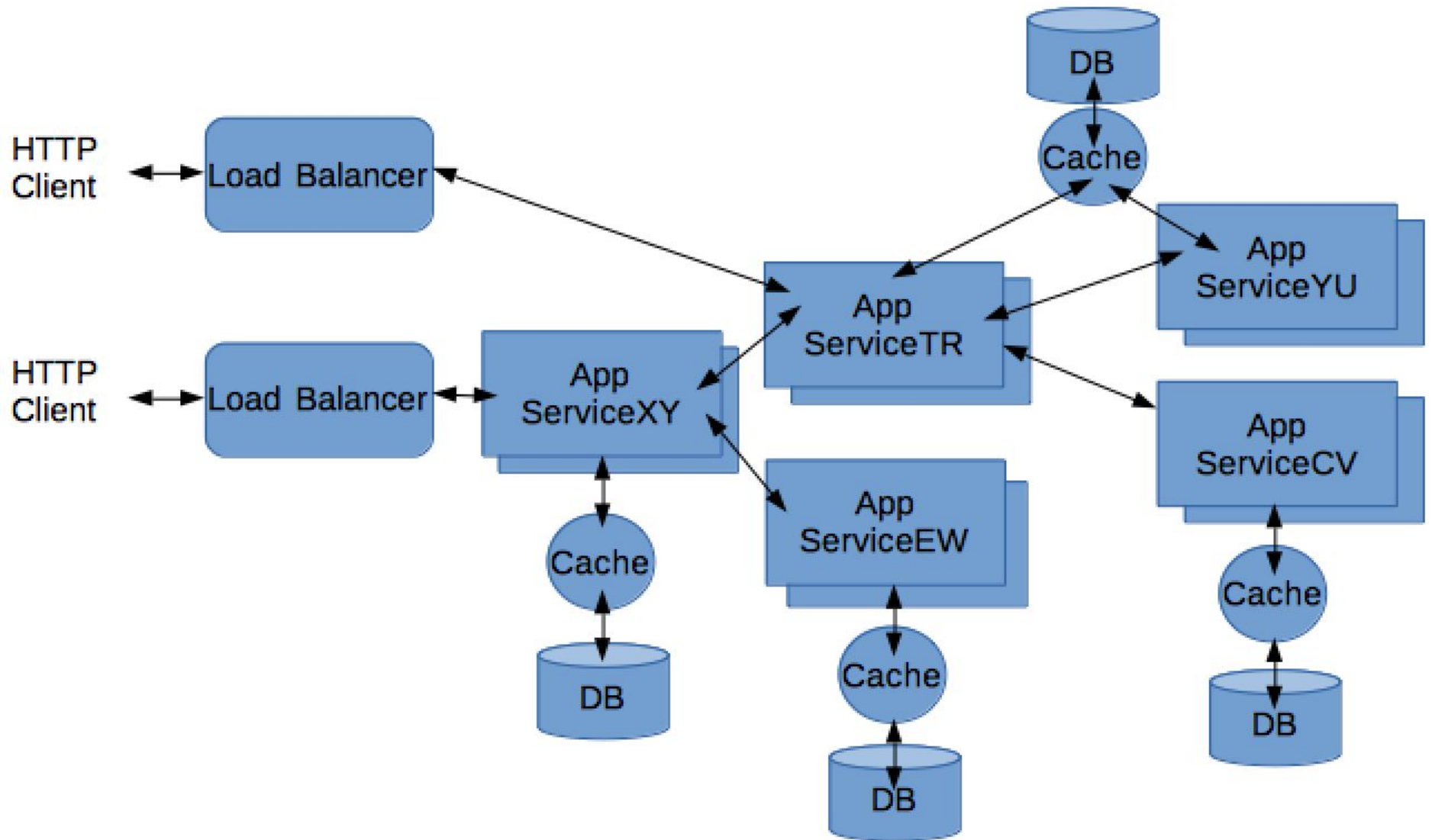
... and scales by distributing these services across servers, replicating as needed.



Key Points

- Distributed services rather than monolith
- Best practice seems to be 1 functional area per service
- No sneaky backdoors, everything goes through APIs and Messaging Services
- No specific technology required!

Some Architectural Patterns



What you used to do

- A monolith where all authentication used to happen.
- Maybe a distributed cache to handle session state

The new fancy way

- You have an authentication service just like all the other services.
- It should be distributed and fault tolerant
- It should be able to scale independently
- It should have a standard way to interact

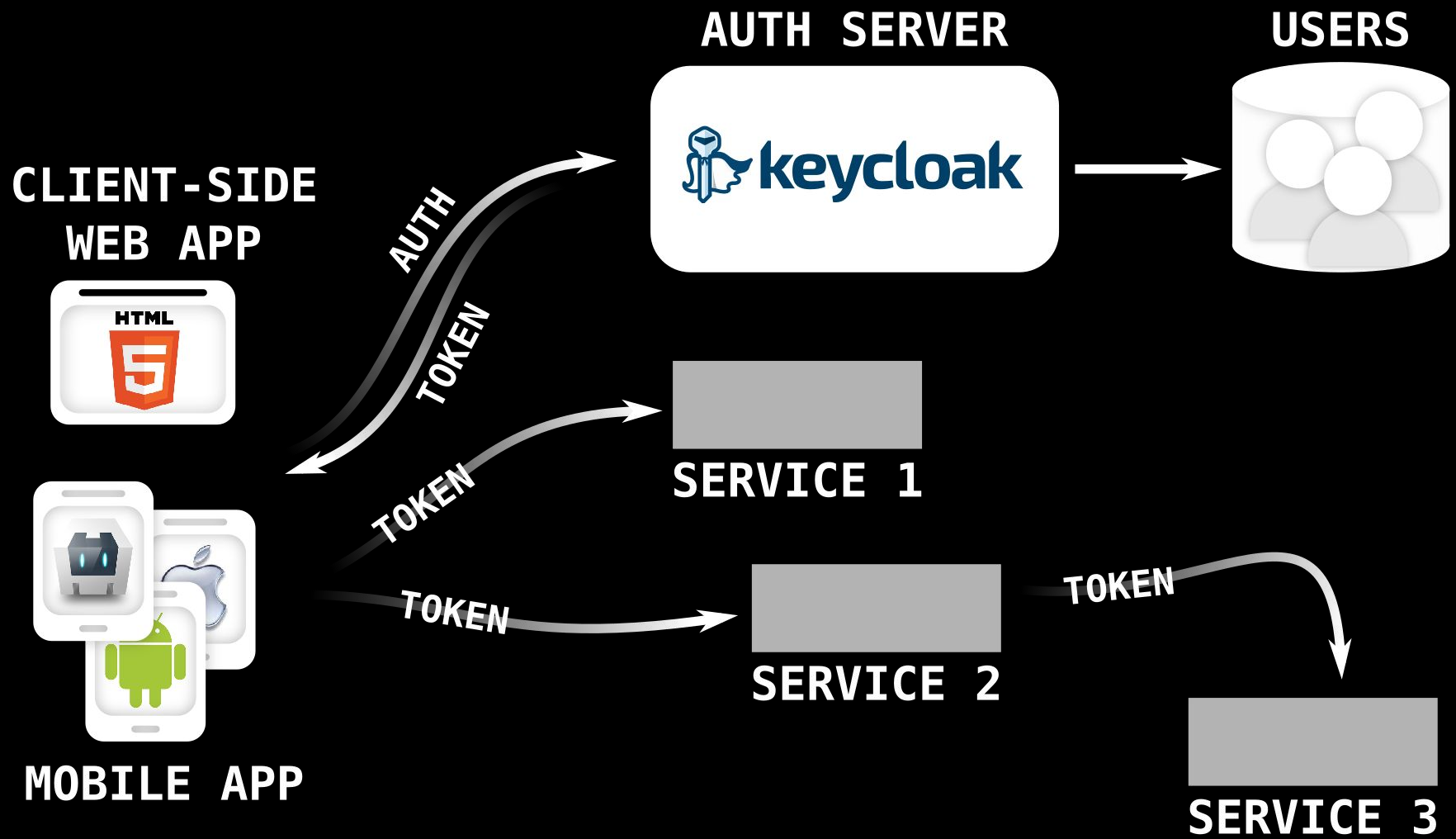
So how do I get my hands on this hotness

- Stormpath
- Auth0
- We are going to show an installable FOSS project from Red Hat called Keycloak



keycloak

Open Source
Identity and Access Management
by Red Hat



Authentication

- Keycloak deals with authentication
- Safely store passwords
- Two factor authentication
- Customizable flows
- Single Sign On


Protocols

- OpenID Connect
- SAML v2
- Kerberos SSO Bridge

Clustering

- Scalability
- High availability
- Based on Infinispan caches

Admin Console

keycloak

Stian Thorgersen ▾

Master ▾

Configure

Realm Settings

Clients

Roles

Identity Providers

User Federation


Authentication

Manage

Users

Sessions


Events

Master 

General Login Keys Email Themes Cache Tokens Security Defenses

* Name


master

Enabled 

ON ☐

Save Cancel

Account Management

 **keycloak**

Back to security-admin-consoleSign Out

Account >

Password

Authenticator

Sessions

Applications

Edit Account

* Required fields

Username

stianst

Email *

stian@redhat.com

First name *

Stian

Last name *

Thorgersen

Cancel

Save

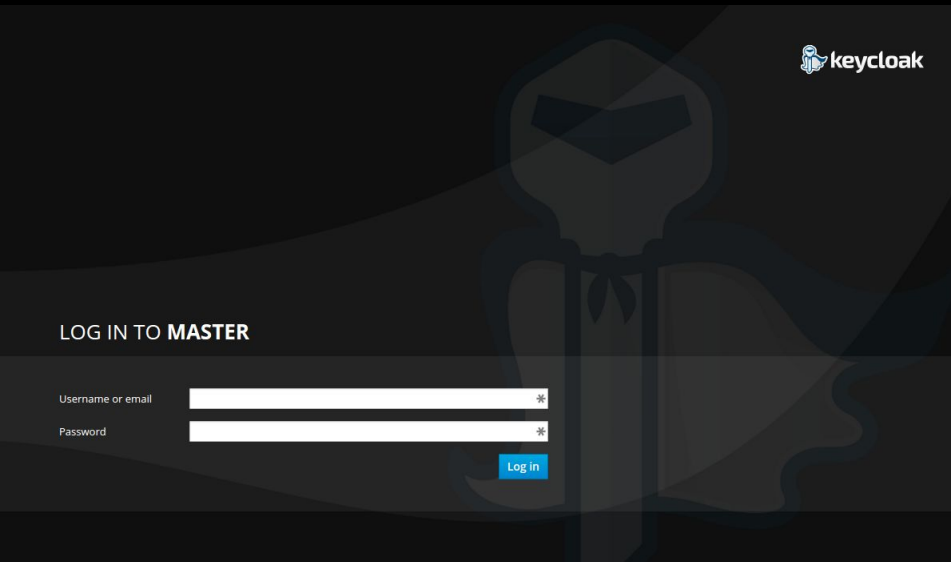
Client Adapters

- Easy to use client libraries
- JavaEE, Spring, JavaScript, NodeJS and mobile
- More to come
- Can also use
 - OpenID Connect Resource Provider libraries
 - SAML v2 Service Provider libraries

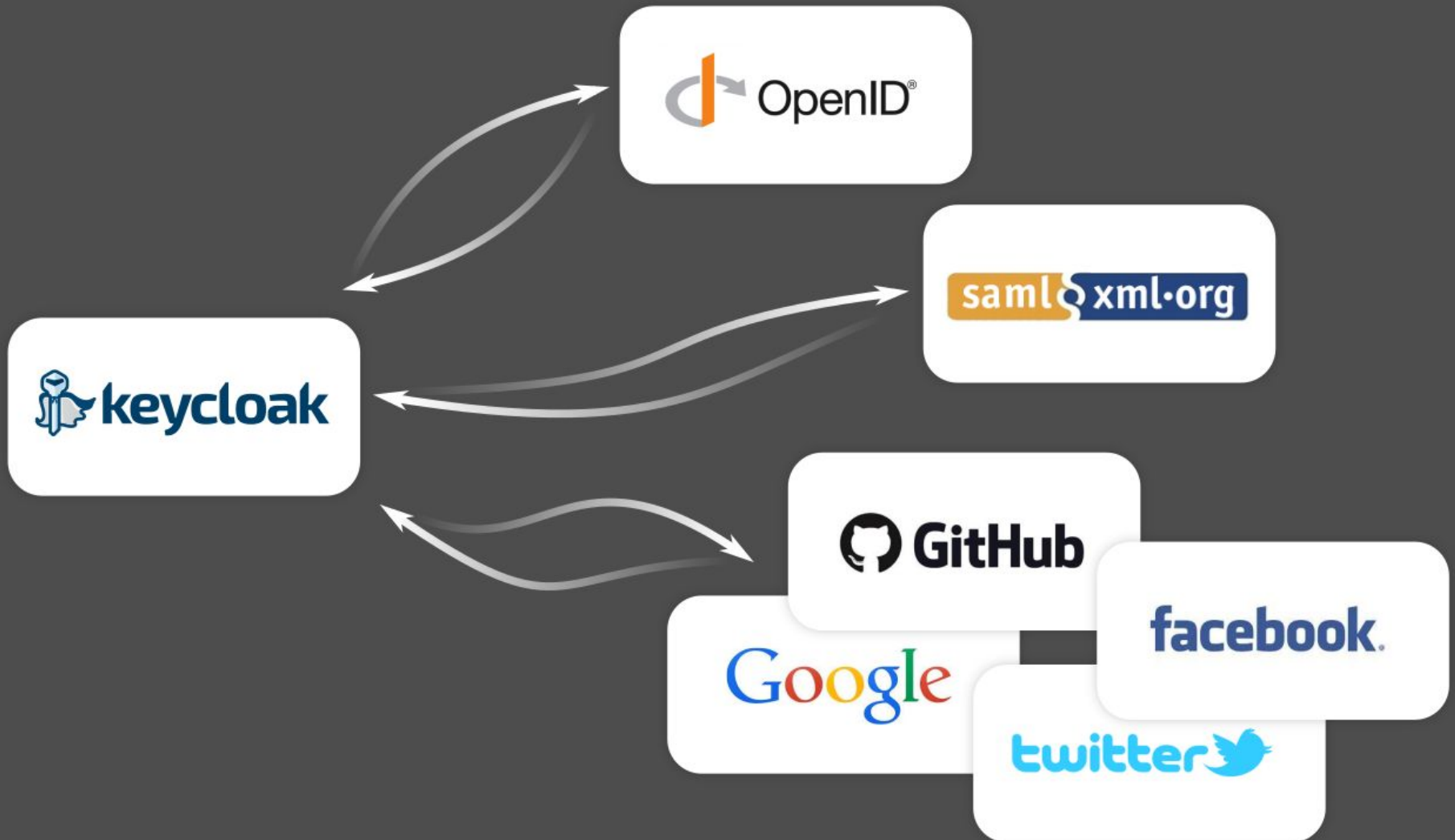
Service Provider Interfaces (SPIs)

- Keycloak aims to be an out of the box solution
- But! Some people have custom requirements
- That's why we have SPIs
 - User Store
 - User Federation
 - Authentication
 - Required Actions
 - And many more, even pluggable protocols

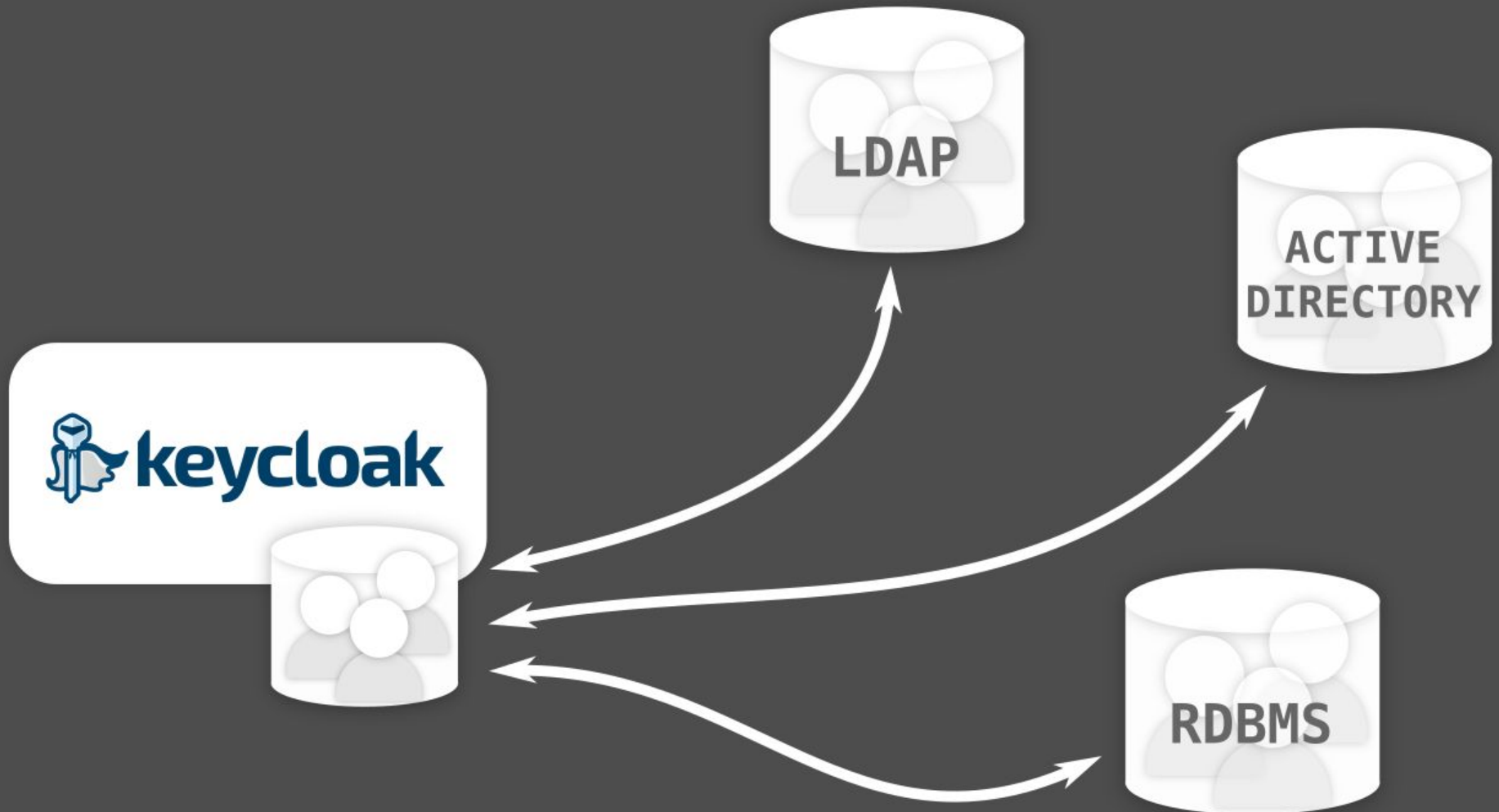
Themes



Identity Brokering



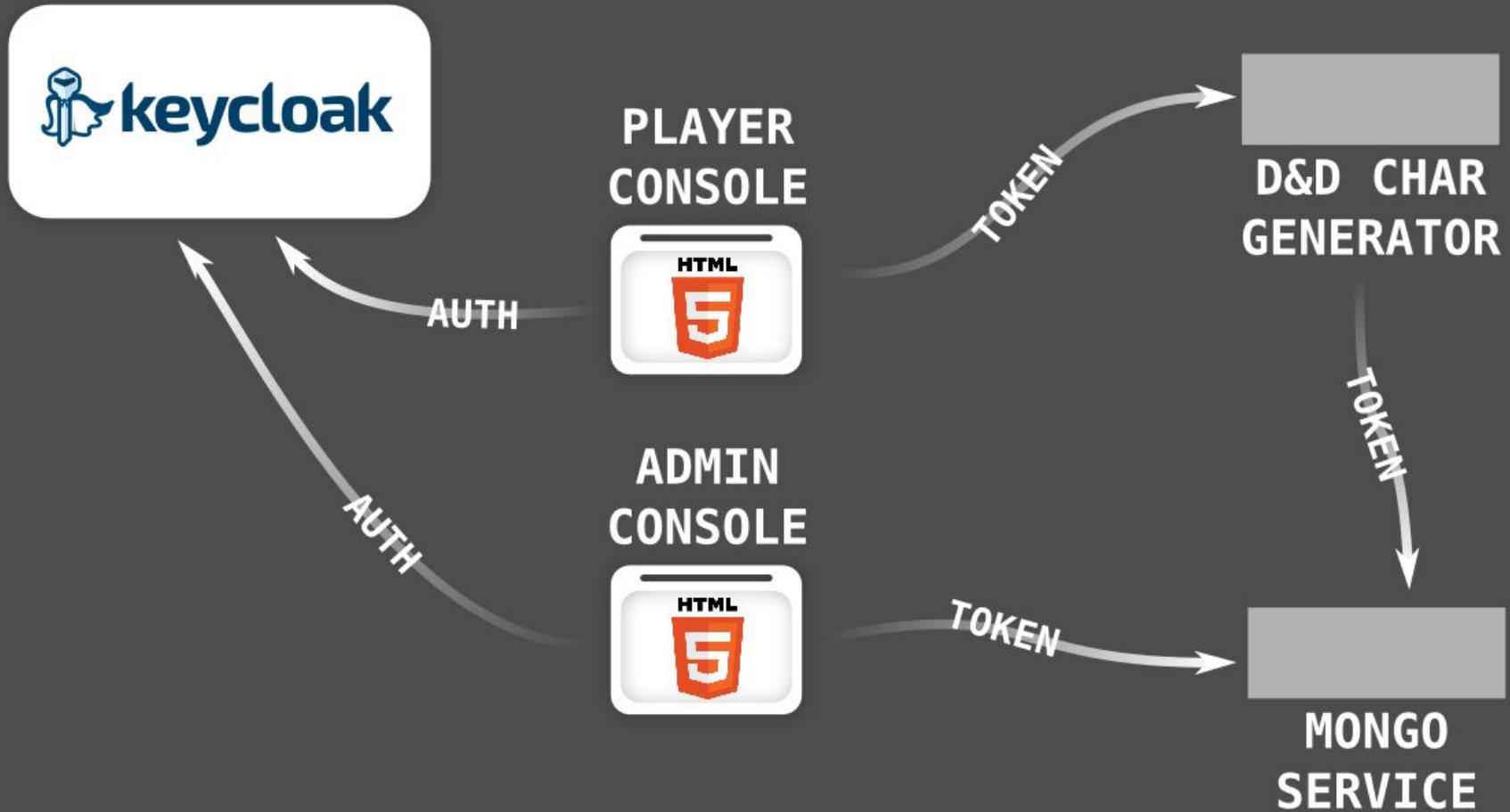
User Federation



Demo Time!

- Take some microservices with different patterns
- Show how easily we added Keycloak
- Show the magic we can now do
- PROFIT!!!

Architecture



<https://bit.ly/j1ddgen>

Take Aways

- Microservices architecture brings some interesting application capabilities
- It also adds some interesting challenges – authentication being a BIG one
- Make your authentication just another service
- Products/projects like Keycloak, Stormpath, and Auth0 can make it easy to bring the authentication love