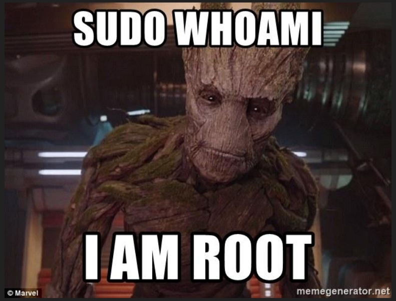
Building secure applications with keycloak (OIDC/JWT)

Abhishek Koserwal
Red Hat

IAAA Security Factor



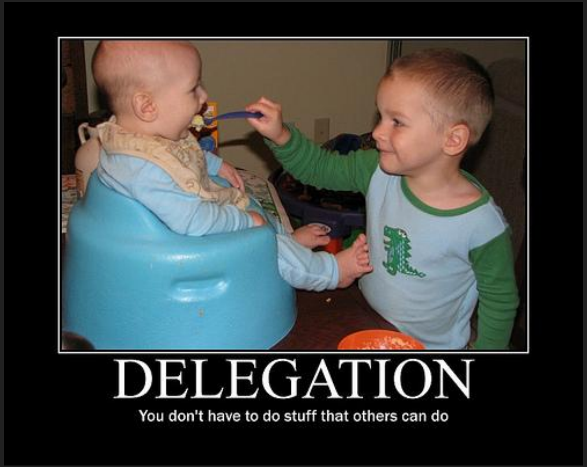
- **Identification**: a set of attributes related to an entity
  - (eg: user -> attribute [name, email, mobile])

- **Authentication**: is the process of verifying an identity
  - (who they say they are)

- **Authorization**: is the process of verifying what someone is allowed to do
  - (permissions)

- **Accounting**: logs, user actions, traceability of actions

Oauth 2 & OpenID Connect



Oauth 2 != Authentication, only Authorization

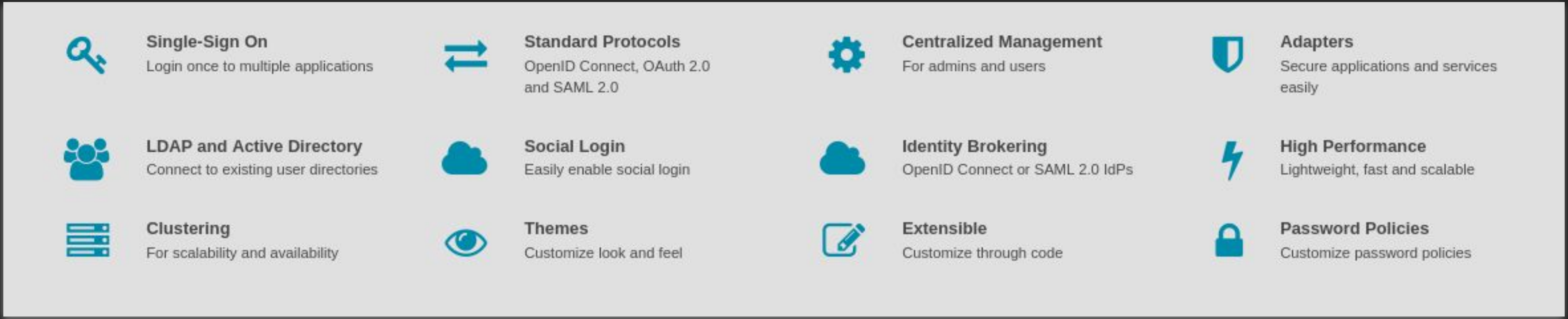
OpenID Connect = Identity + Authentication + Authorization

50+ Security Specifications...

What is Keycloak?

Open Source

Identity Solution for Applications, Services and APIs



Single-Sign On
Login once to multiple applications

Standard Protocols
OpenID Connect, OAuth 2.0 and SAML 2.0

Centralized Management
For admins and users

Adapters
Secure applications and services easily

LDAP and Active Directory
Connect to existing user directories

Social Login
Easily enable social login

Identity Brokering
OpenID Connect or SAML 2.0 IdPs

High Performance
Lightweight, fast and scalable

Clustering
For scalability and availability

Themes
Customize look and feel

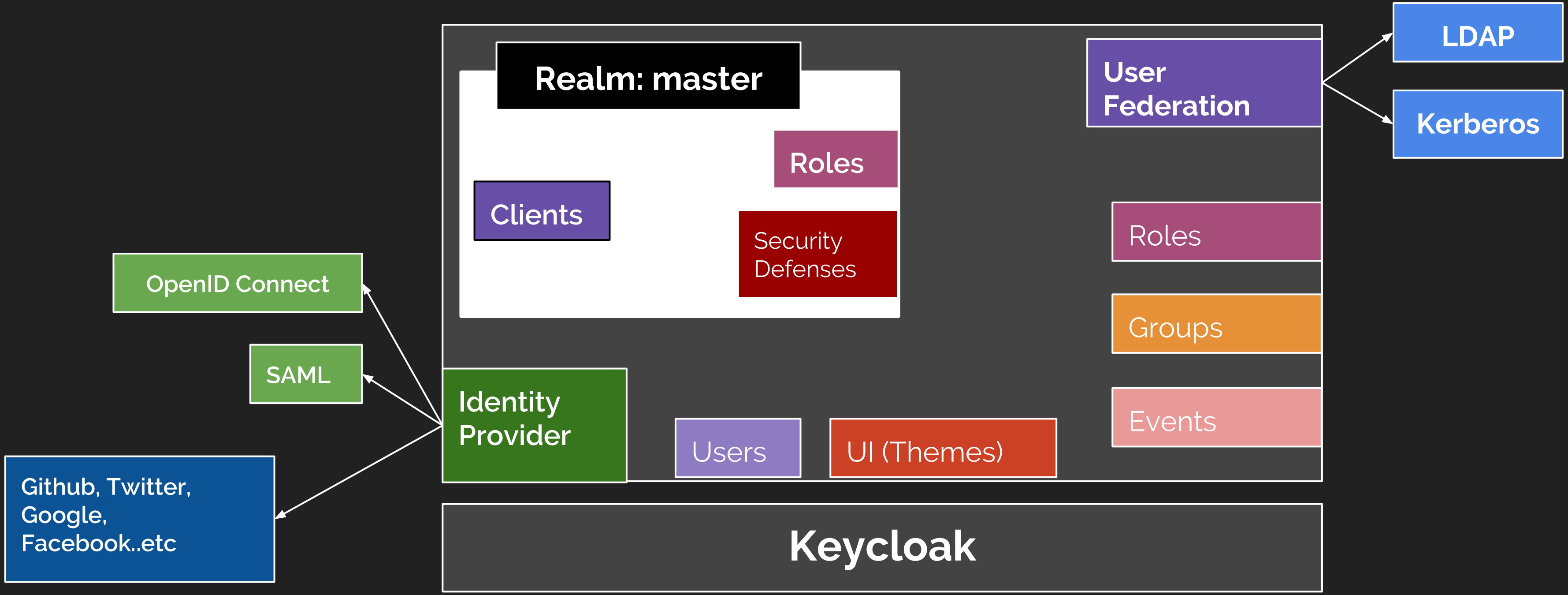
Extensible
Customize through code

Password Policies
Customize password policies

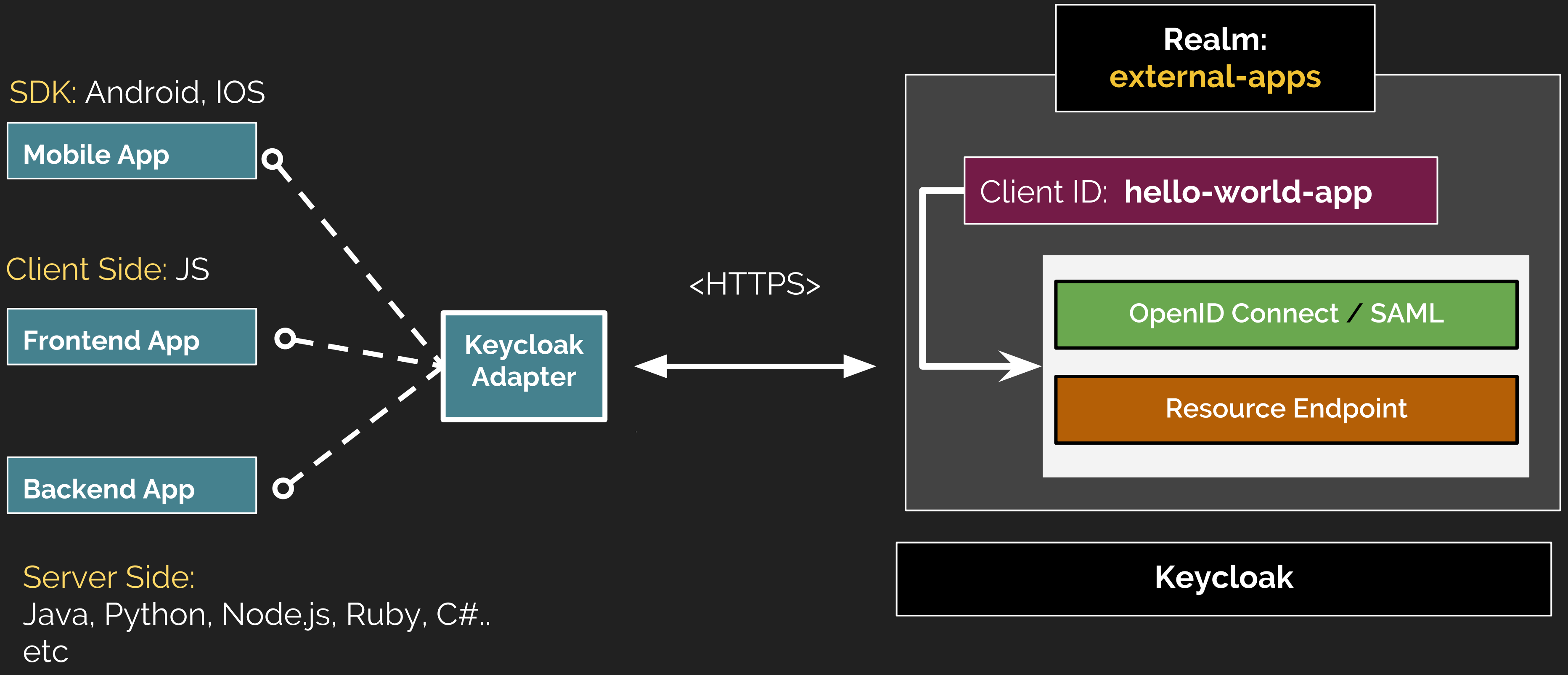
Why to use keycloak?

- **Reliable Solution**
- **! Reinventing the wheel ?** (shared libraries, keys/certs, configuration, standards)
- **Open Source (3C's)**
    - **Cost**
    - **Customizable / Contributions**
    - **Community**
- **Hybrid Cloud Model**

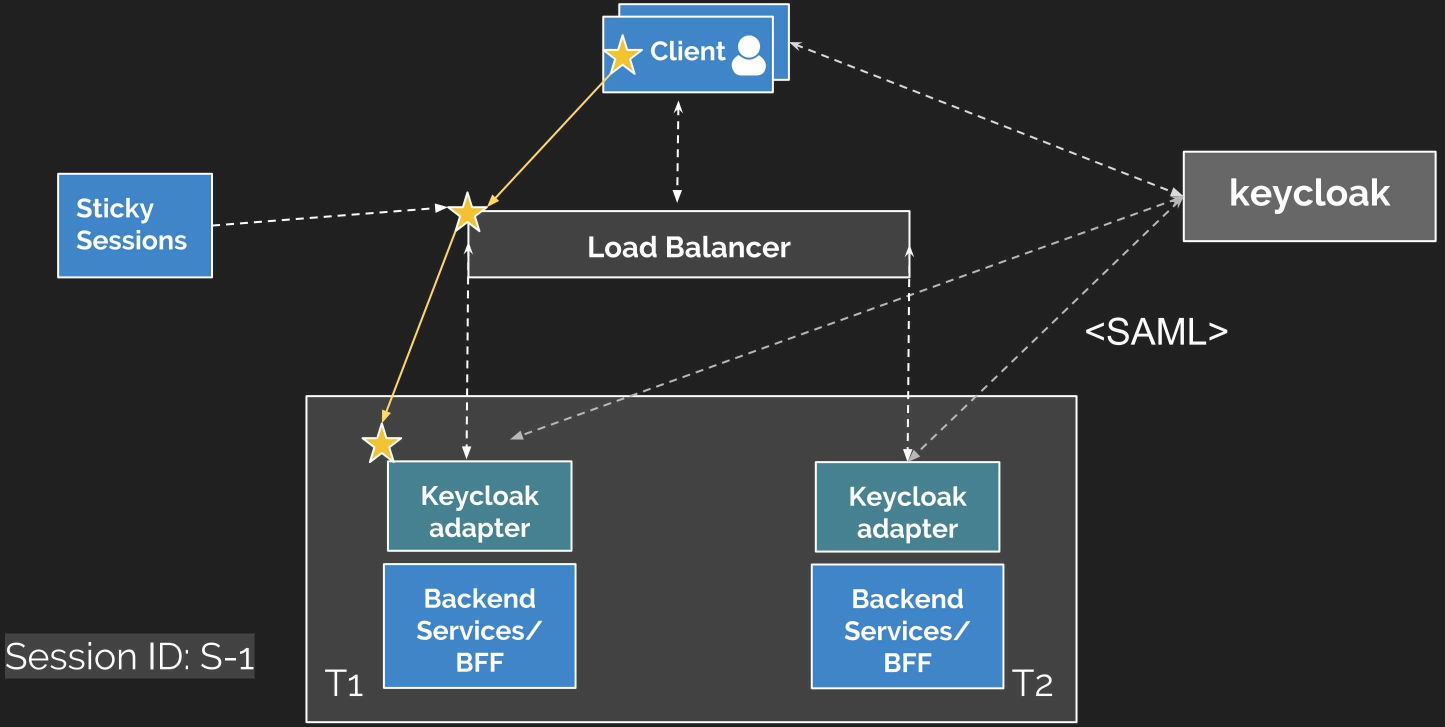
Core Concepts



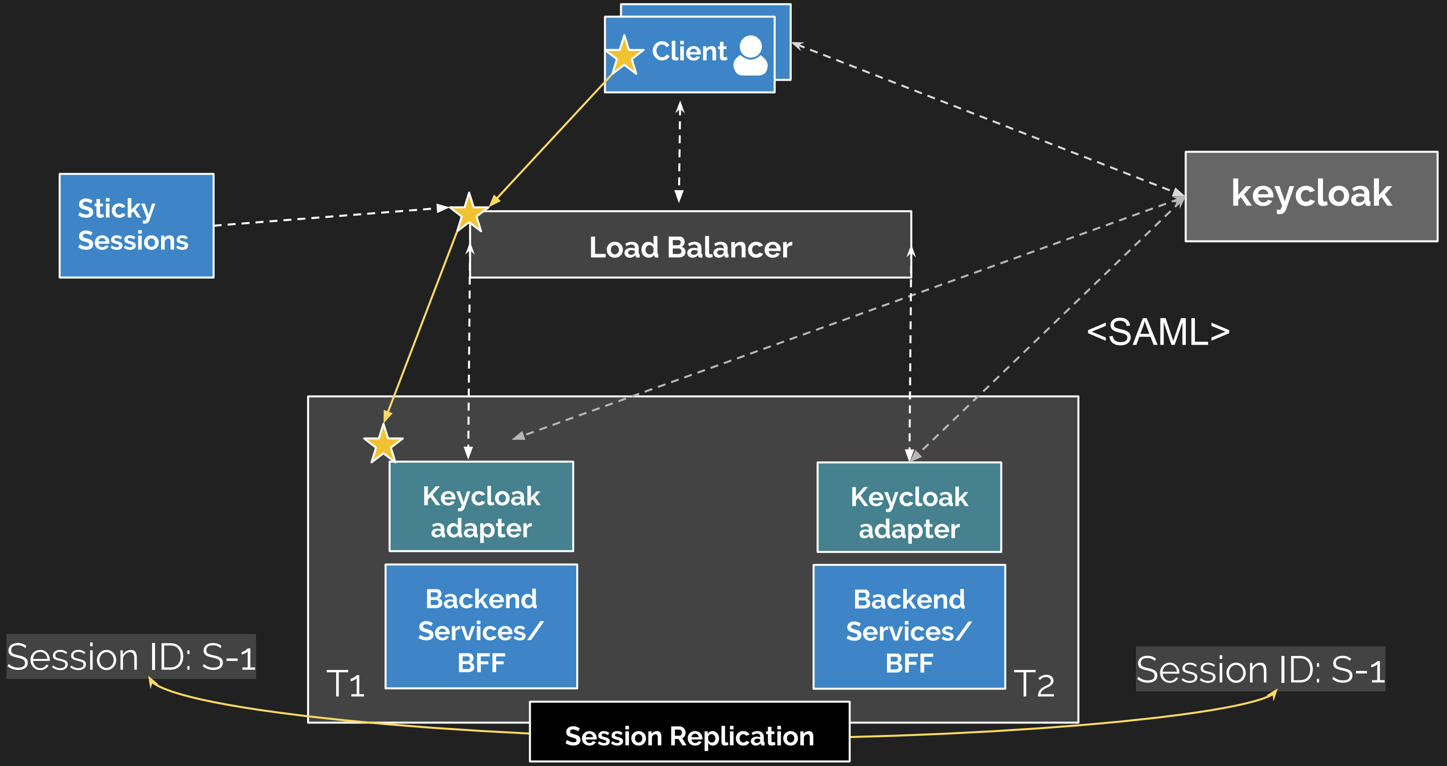
App: Integration



How we used..


Client
Sticky Sessions
Load Balancer
keycloak
<SAML>
Keycloak adapter
Backend Services/ BFF
Keycloak adapter
Backend Services/ BFF
T1
T2
Session ID: S-1

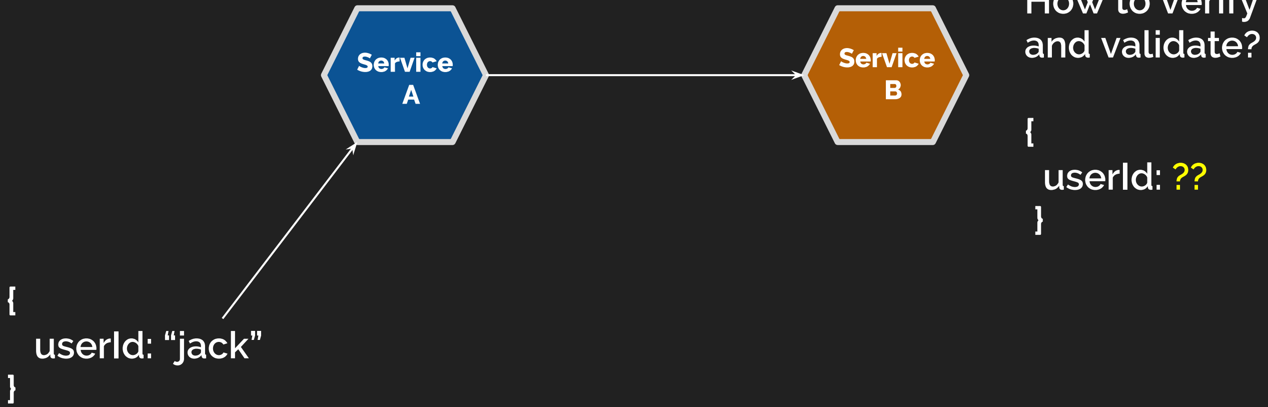
How we used..


Client
Sticky Sessions
Load Balancer
keycloak
<SAML>
Keycloak adapter
Backend Services/ BFF
Keycloak adapter
Backend Services/ BFF
T1
T2
Session ID: S-1
Session ID: S-1
Session Replication

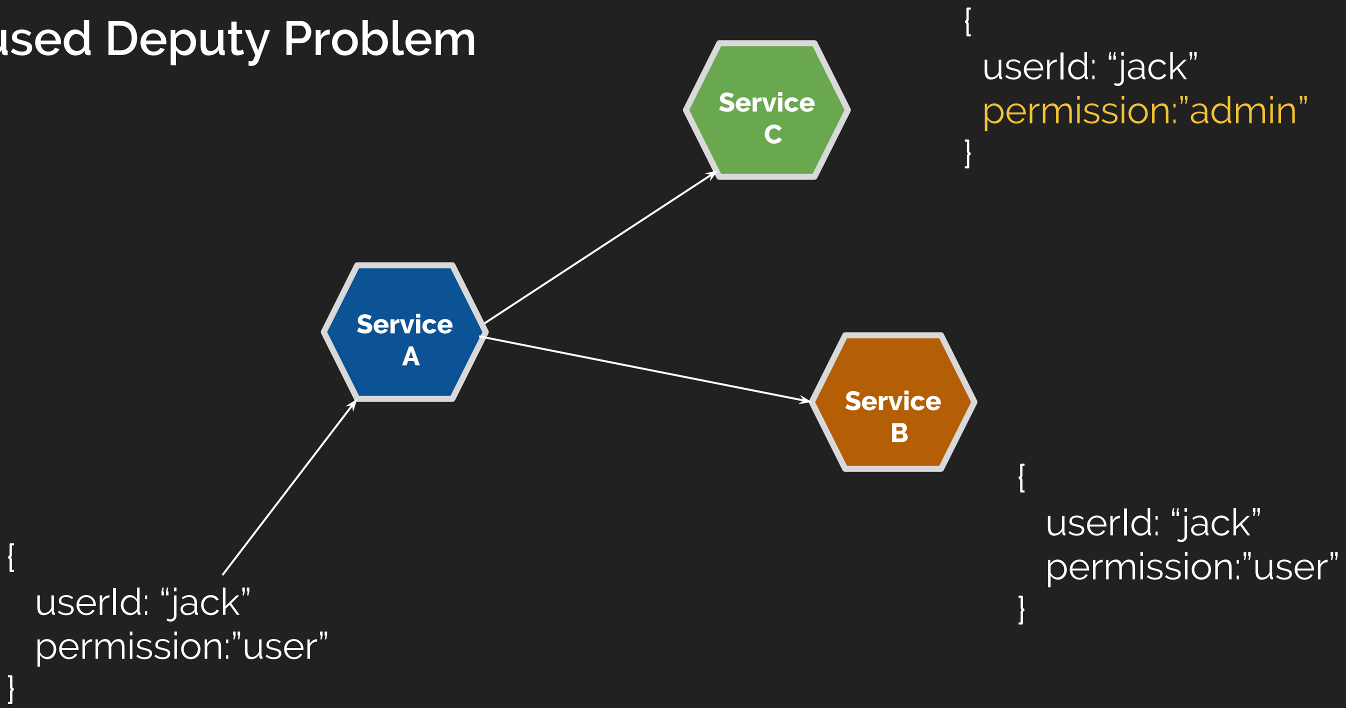
Problems

- Scalability with server side sessions
- Sticky Sessions are Evil
- Shifting monolith to Openshift/Containers (stateful -> stateless)

Service-to-Service : Authentication & Authorization



The Confused Deputy Problem

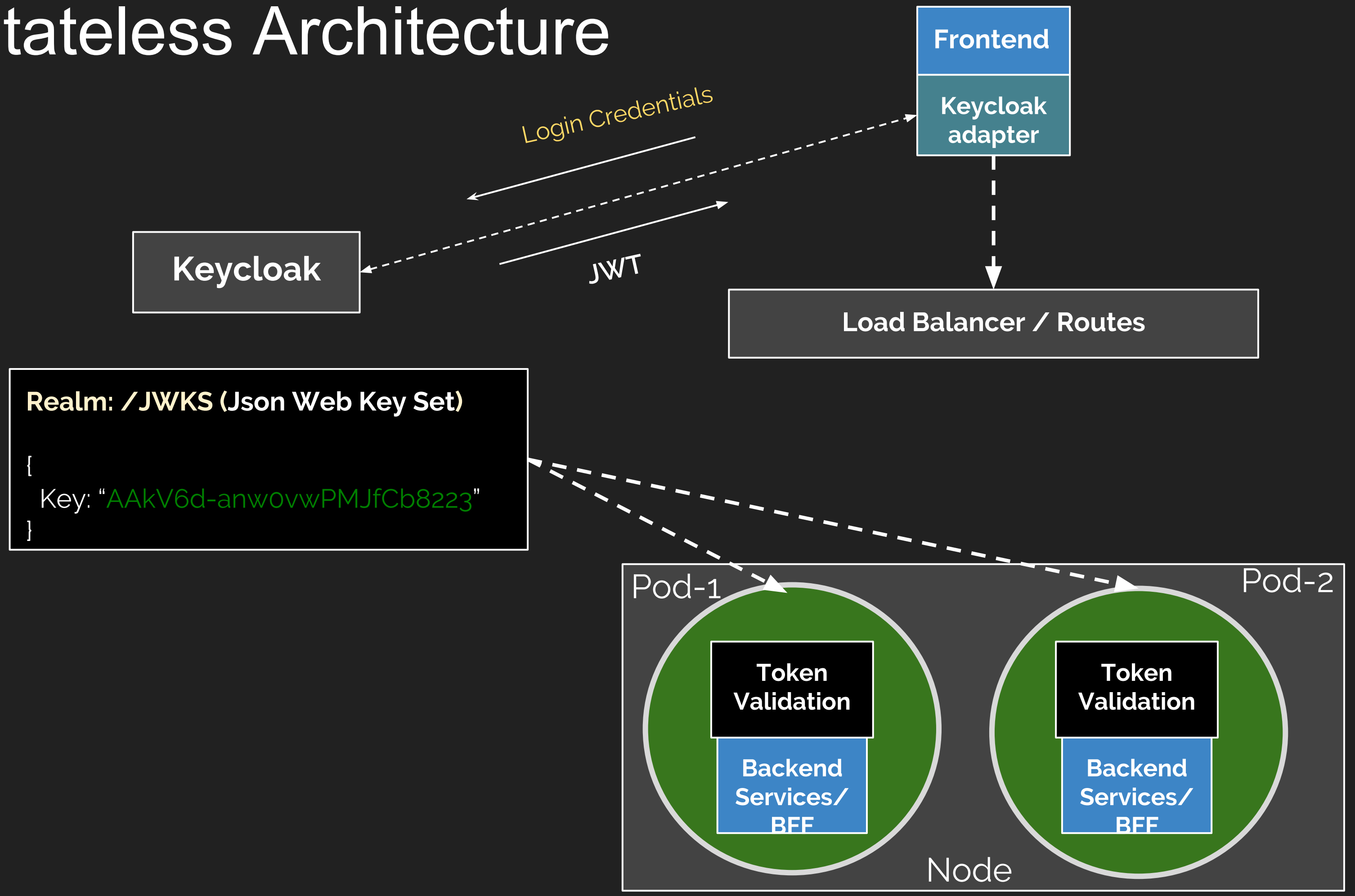


```
{
  userId: “jack”
  permission:”admin”
}
```

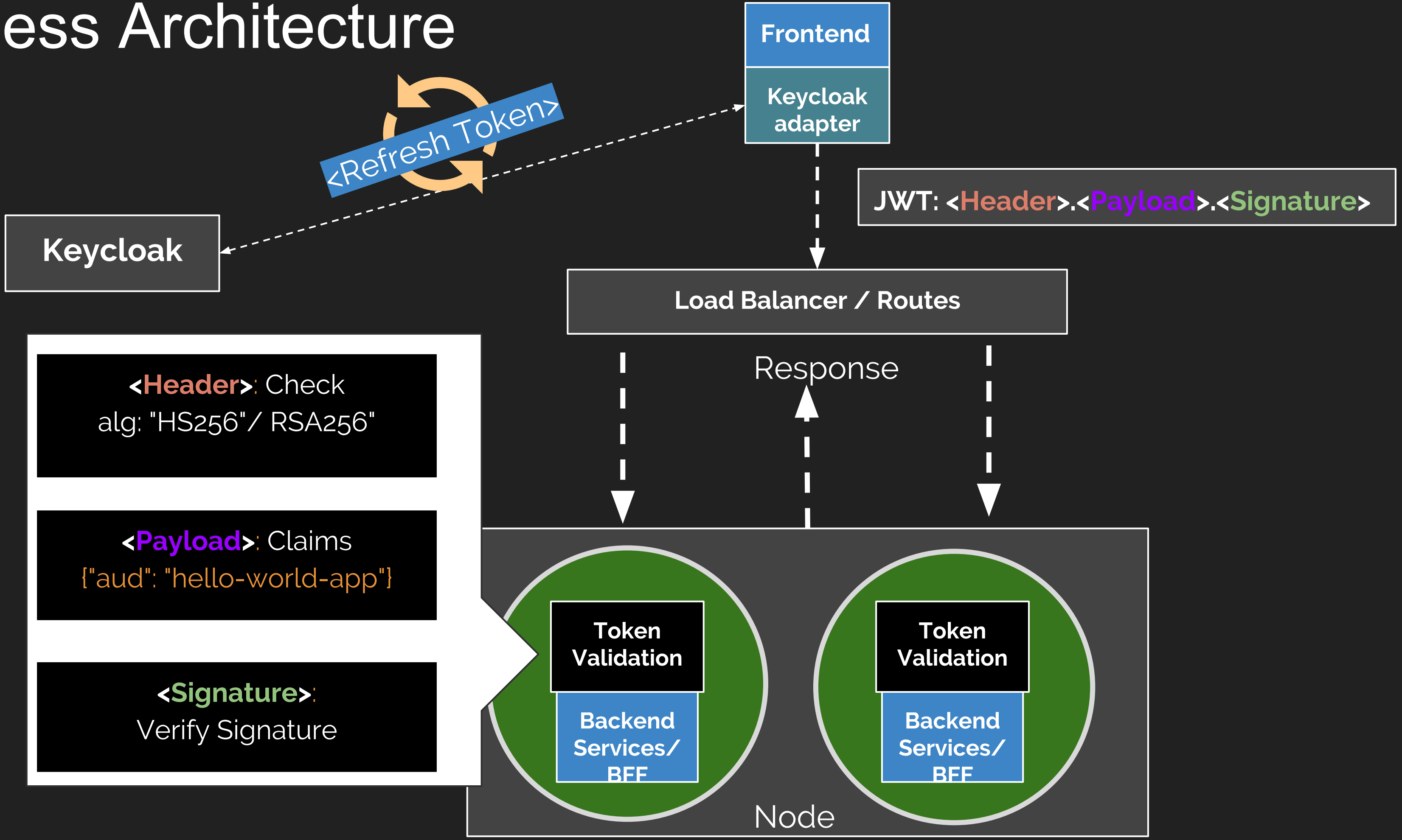
```
{
  userId: “jack”
  permission:”user”
}
```

```
{
  userId: “jack”
  permission:”user”
}
```

Stateless Architecture



Stateless Architecture



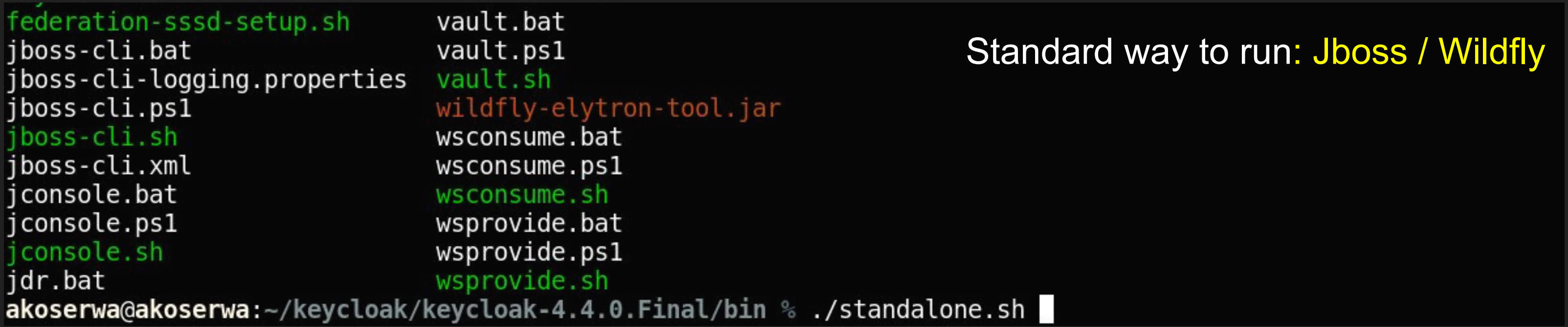
Setup: keycloak

Require docker daemon running

```
docker pull jboss/keycloak

docker run -d -e KEYCLOAK_USER=<USERNAME> -e KEYCLOAK_PASSWORD=<PASSWORD> -p 8081:8080 jboss/keycloak
```

Standalone server distribution
(https://www.keycloak.org/downloads.htm)



```
federation-sssd-setup.sh       vault.bat
jboss-cli.bat                  vault.ps1
jboss-cli-logging.properties   vault.sh
jboss-cli.ps1                  wildfly-elytron-tool.jar
jboss-cli.sh                   wsconsume.bat
jboss-cli.xml                  wsconsume.ps1
jconsole.bat                   wsconsume.sh
jconsole.ps1                   wsprovide.bat
jconsole.sh                    wsprovide.ps1
jdr.bat                        wsprovide.sh
akoserwa@akoserwa:~/keycloak/keycloak-4.4.0.Final/bin % ./standalone.sh
```

Standard way to run: Jboss / Wildfly

Application Demo

JWT: Json Web Tokens

- JWT over HTTPS and never HTTP
- **Access tokens:** are tokens that give those who have them access to protected resources (Short lived)
- **Refresh tokens:** allow clients to request new access tokens.
- Cookie vs local storage
  - local storage prone to cross-site scripting (XSS)
  - Cookie only with HttpOnly flag (size < 4 kb), prone to Cross-Site Request Forgery (CSRF)

Keycloak vs Others

- Designed as a single product
- Easy to setup & configure
- Supports Docker registry Auth
- OpenJDK support
- spring-boot support :

  http://start.spring.io/

Securing keycloak

- Make sure to secure keycloak end-points
- IP Restriction/Port restriction for the endpoint/auth/admin console
- Configure security defenses like: Password guess: brute force attacks
- If an access token or refresh token is compromised, revocation policy to all applications
- Client config: hostname is based on the request headers.

Q & A

Thank You!