

Best practices for creating
secure microservices

Erin Schnabel
@ebullientworks

InterConnect
2017

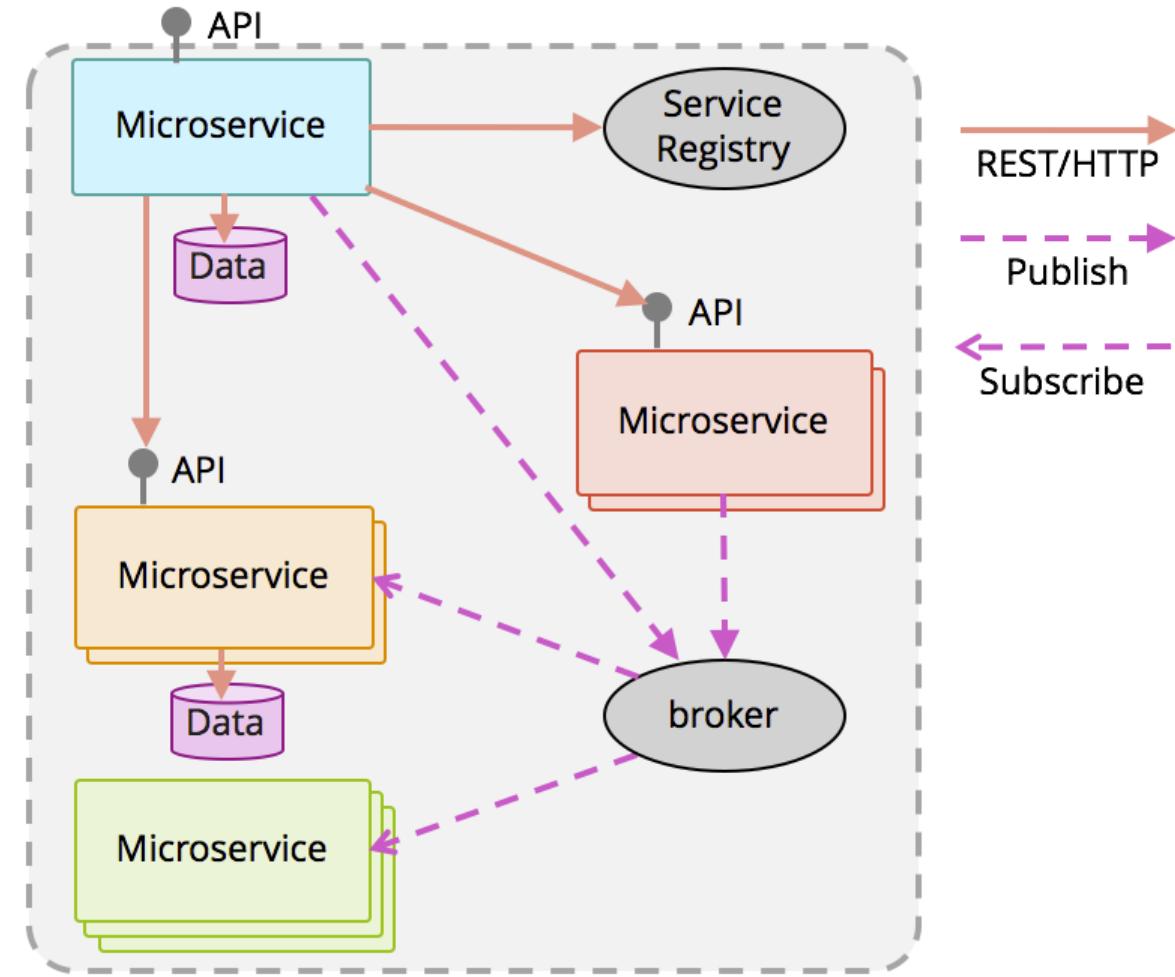


Agenda

- Microservices
- Security Challenges
- Technologies
- Examples

Microservices are used to...

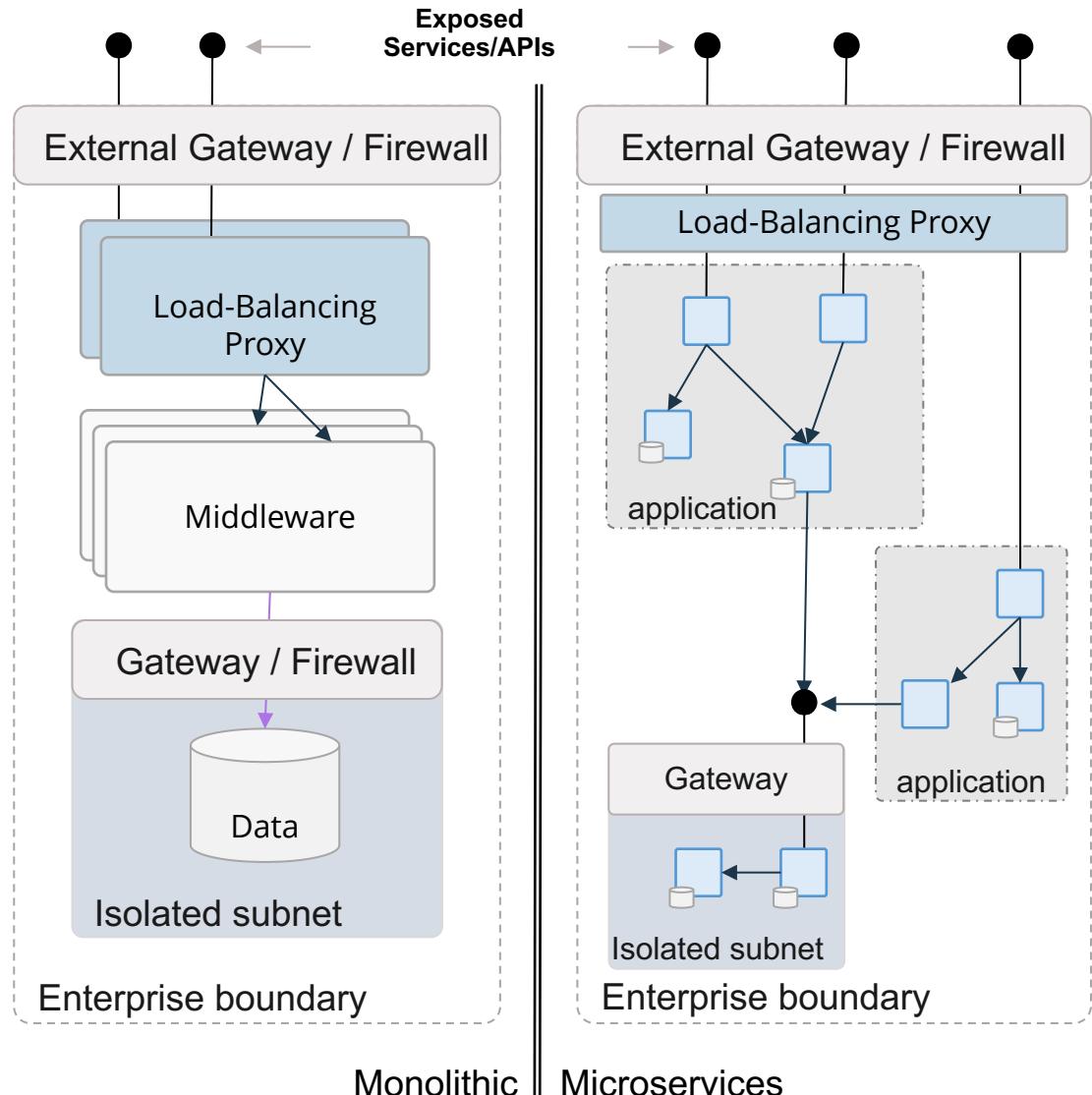
- compose a *complex application* using
 - “small”
 - independent (autonomous)
 - replaceable
 - processes
- that communicate via
 - language-agnostic APIs



Security Challenges

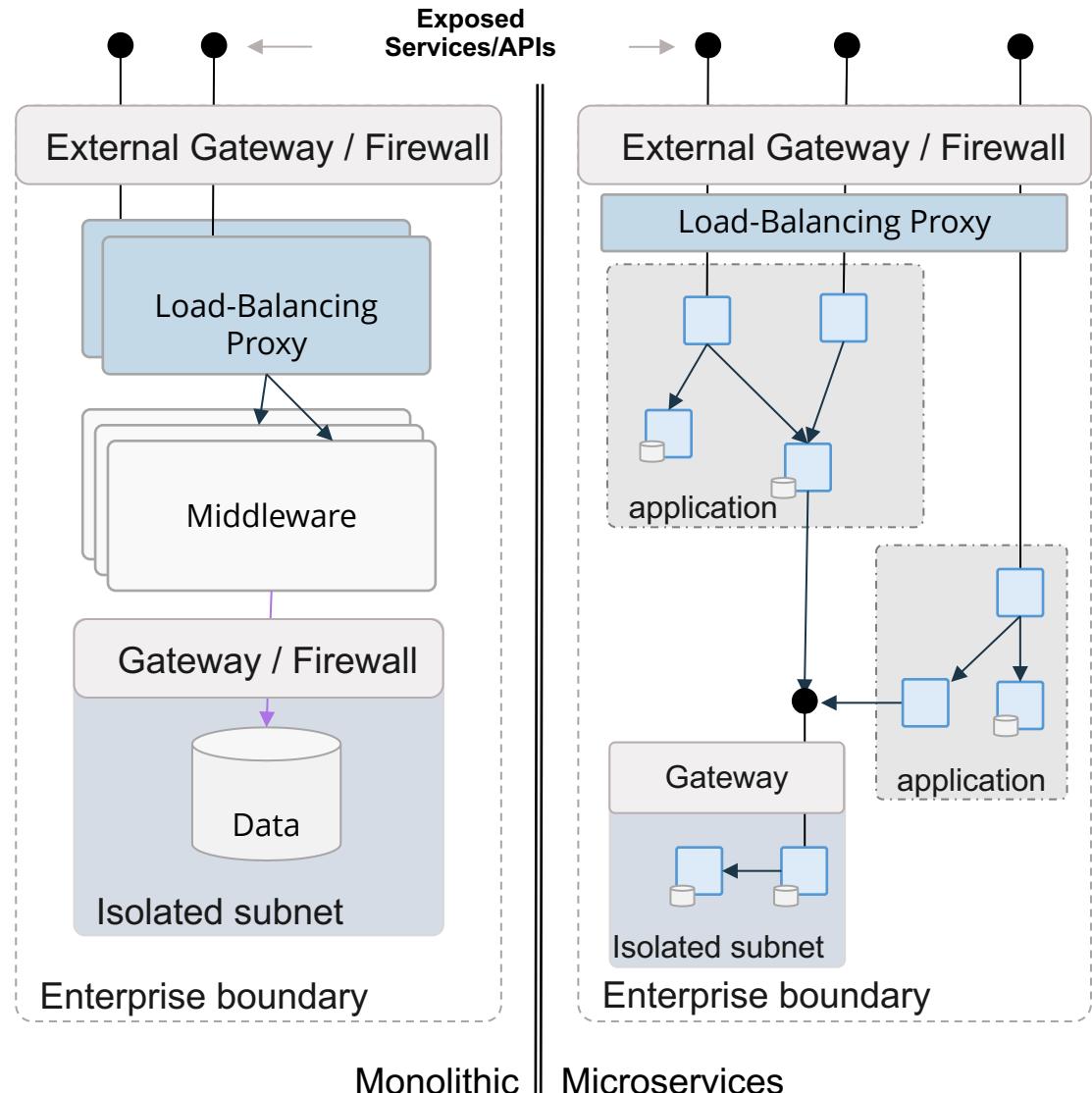
API Surface Explosion

- Application composed of services
- Each service with its own API
- Each service with its own security concerns.



Service to Service

- Services care about originating ID
- Services need to establish trust
 - Who is invoker?
- Avoid bottlenecks
 - E.g. calls to centralized service



Identity, authorization and authentication are not piece of cake

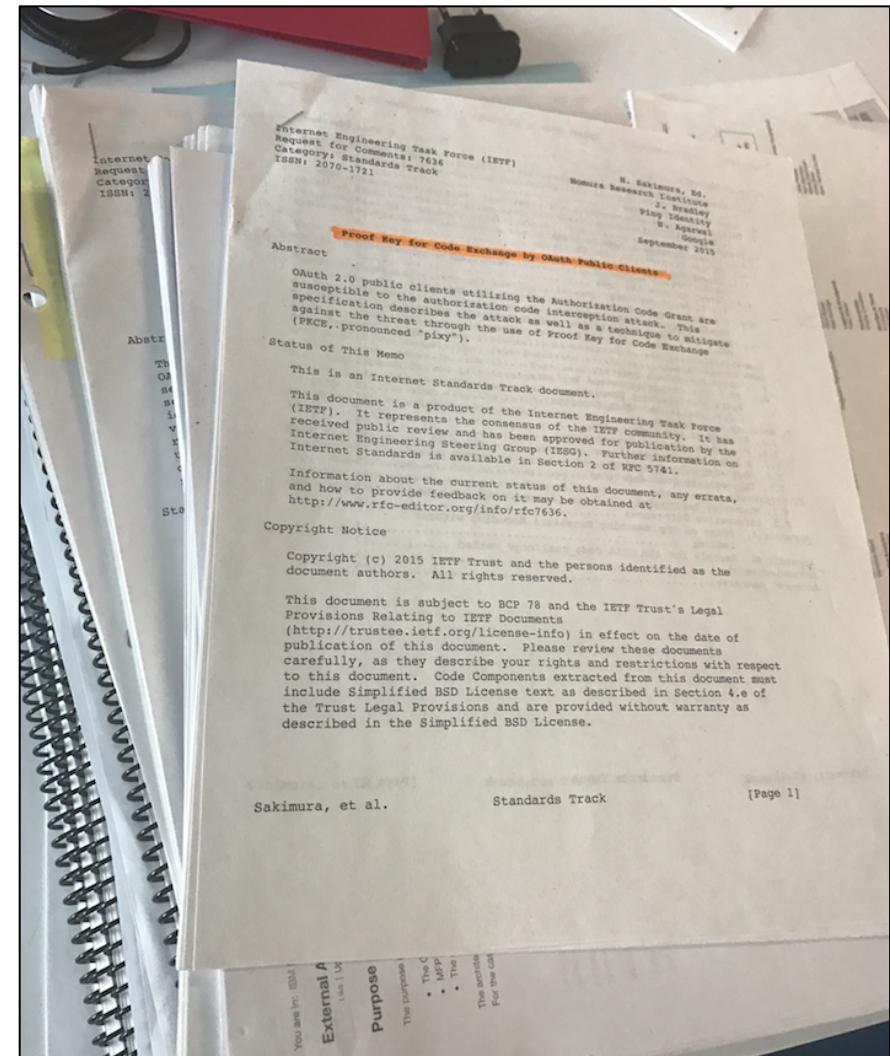


!=

A large black exclamation mark followed by a large black equals sign, positioned centrally between the two images.

Care to read some RFCs?

1. [RFC 6749](#) - The OAuth 2.0 Authorization Framework
2. [RFC 6750](#) - The OAuth 2.0 Authorization Framework: Bearer Token Usage
3. [Open ID Connect](#)
 1. [Core](#)
 2. [Discovery](#)
 3. [Dynamic registration](#)
4. [RFC 7591](#) - OAuth 2.0 Dynamic Client Registration Protocol
5. [RFC 7592](#) - OAuth 2.0 Dynamic Client Registration Management Protocol
6. [RFC 7518](#) - JSON Web Algorithm (JWA)
7. [RFC 7519](#) - JSON Web Token (JWT)
8. [RFC 7515](#) - JSON Web Signature (JWS)
9. [RFC 7636](#) - Proof Key for Code Exchange by OAuth Public Clients
10. [RFC 7662](#) - OAuth 2.0 Token Introspection
11. [RFC 7521](#) - Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants
12. [RFC 7523](#) - JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants
13. [RFC 7522](#) - Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants
14. [RFC 7642](#) - System for Cross-domain Identity Management: Definitions, Overview, Concepts, and Requirements
15. [RFC 7643](#) - System for Cross-domain Identity Management: Core Schema
16. [RFC 7644](#) - System for Cross-domain Identity Management: Protocol



Everything revolves around two standards - OAuth2 and OIDC



Authorization



Authentication

OAuth2 / OIDC

- Authentication Server
- Used by most Social Login today
 - (Facebook, Google, GitHub etc)
- Auth request by User results in Access Token
- Access Token supplied by User as part of secured requests
- Access Token validated by invoked Service querying OAuth server.
 - Allows for token invalidation
 - Consider token invalidation mid-request impact on overall request flow.
- Open ID Connect (OIDC) as an identity layer over OAuth2

JWT

- JSON Web Tokens
- JSON Claims + Digital Signature Block
 - Base64 Encoded for transmission.
- Can be encrypted, but usually not.
- Signature can be signed with a Shared Secret, or Public/Private Key.
- Standard time related fields restrict validity period of JWT
- Self Verifying. No need to call external server.

JWK

- JSON Web Key

- Similar to JWT:

<u>4.</u>	JSON Web Key (JWK) Format	<u>5</u>
<u> 4.1.</u>	"kty" (Key Type) Parameter	<u>6</u>
<u> 4.2.</u>	"use" (Public Key Use) Parameter	<u>6</u>
<u> 4.3.</u>	"key_ops" (Key Operations) Parameter	<u>7</u>
<u> 4.4.</u>	"alg" (Algorithm) Parameter	<u>8</u>
<u> 4.5.</u>	"kid" (Key ID) Parameter	<u>8</u>
<u> 4.6.</u>	"x5u" (X.509 URL) Parameter	<u>8</u>
<u> 4.7.</u>	"x5c" (X.509 Certificate Chain) Parameter	<u>9</u>
<u> 4.8.</u>	"x5t" (X.509 Certificate SHA-1 Thumbprint) Parameter . . .	<u>9</u>
4.9.	"x5t#S256" (X.509 Certificate SHA-256 Thumbprint)	.

- Makes validating with PKI across languages deterministic
 - No obscurity

API Key / Shared Secret

- Common for WebHooks, User “Apps” (Facebook, Google, etc)
- Establish trust because both parties know the same Key/Secret
- Identity can be established in parallel.
 - Request transmits an ID along with a Key or Secret
 - Essentially UserID & Password
 - Request signs something including the ID using the Key / Secret
 - ID is used to recreate signature, verifying the sender knew ID & Secret, establishing trust.
 - Can be used to further protect additional request content, Eg AWS Auth.

The premise ...

- Hands on with microservices
- Stick with 'Hello World' simplicity
- Choose your own adventure
- Fast path to the hard stuff
- Build something cool (to you!)
- Learn as you go



GAMEON

A Throwback Adventure

You are in a maze of little interconnected rooms, none alike. And you aren't alone...

ENTER

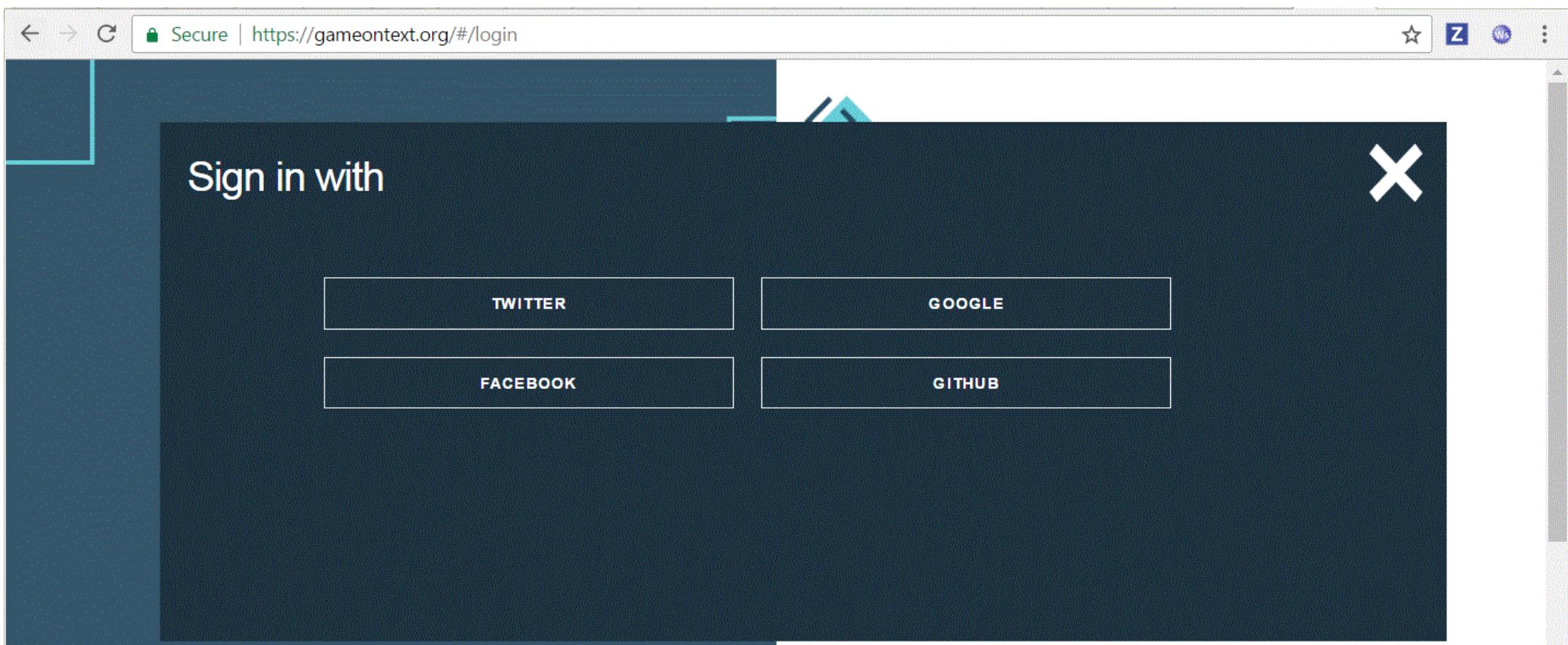


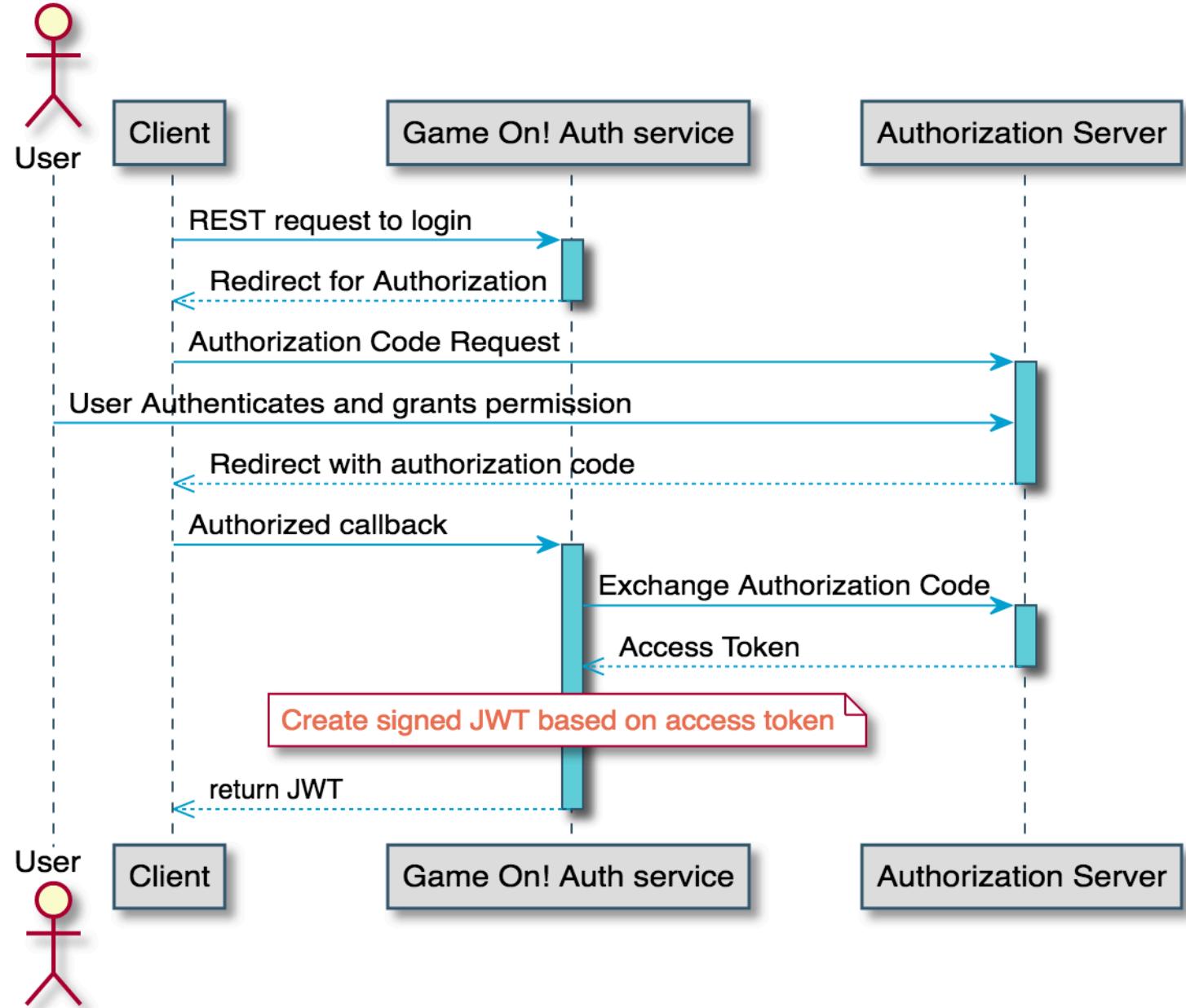
the [wasdev](#) team

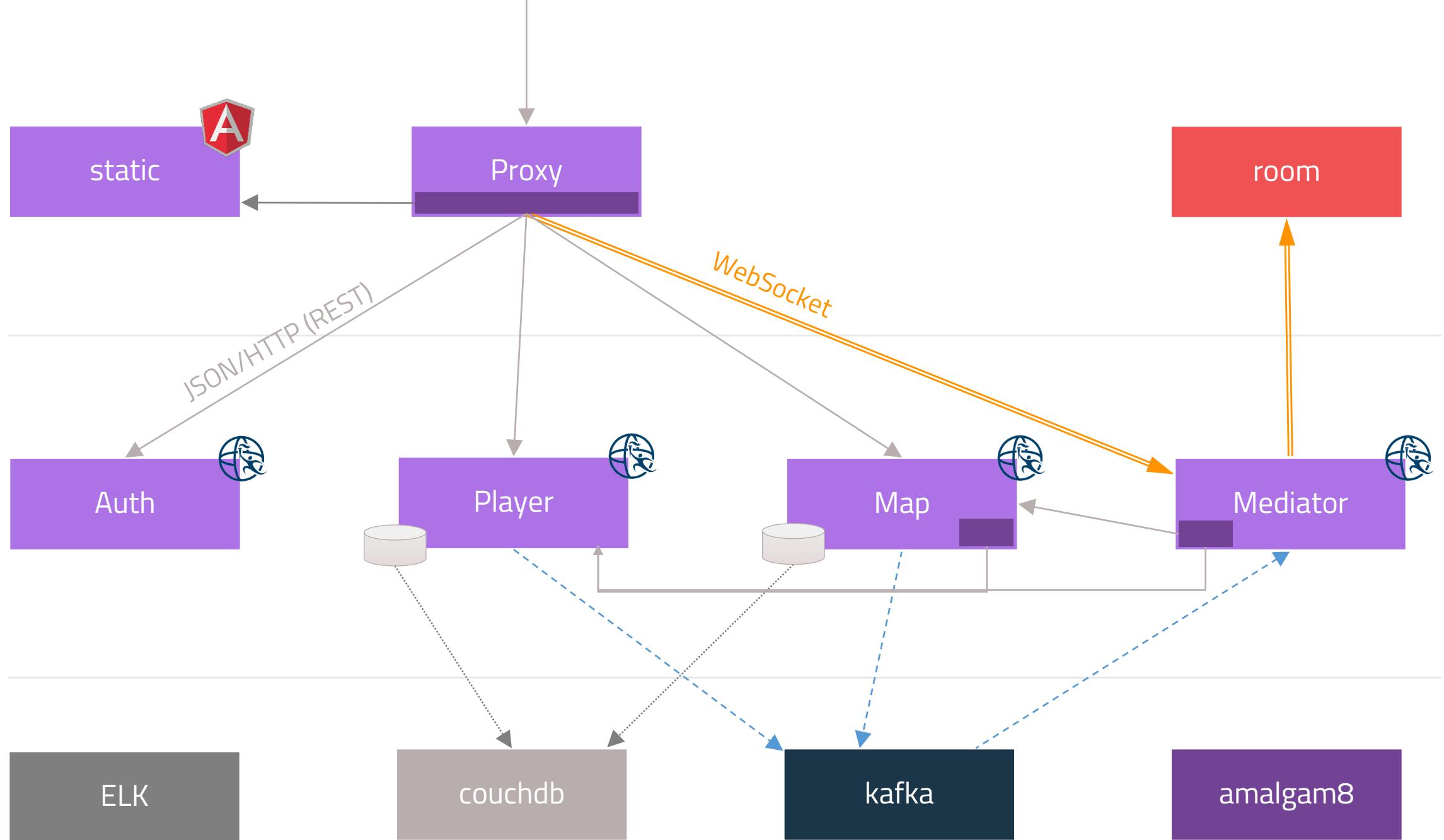
Identifying users

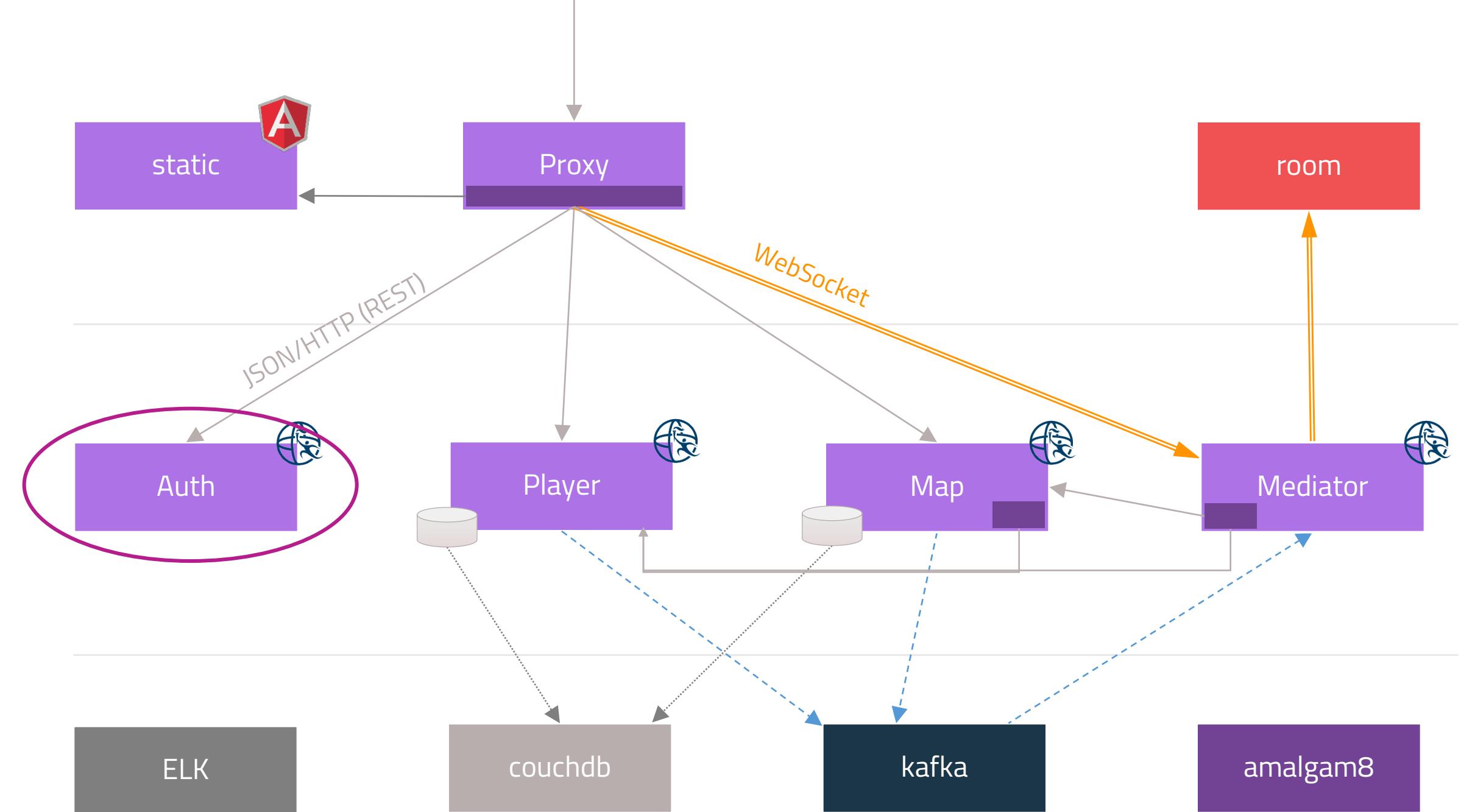
Manage username and password

Social sign-on









JWTs in Game On

- OAuth authenticates & identifies user
- User identity embedded in JWT issued back to browser
- JWT is signed with gameontext.org certificate
- System services deal with JWT
 - validate JWT & trust embedded identity
- Tricky: Signed JWT as a the Random State string during OAuth

```
public String createJwt(Key key) throws Exception {  
    // create and sign the JWT, including a hint  
    // for the key used to sign the request (kid)  
    String newJwt = Jwts.builder()  
        .setHeaderParam("kid", "meaningfulName")  
        .setSubject("user-12345")  
        .setAudience("user")  
        .setIssuedAt(Date.from(Instant.now()))  
        .setExpiration(Date.from(Instant.now().plus(15, ChronoUnit.MINUTES)))  
        .signWith(SignatureAlgorithm.RS256, key)  
        .compact();  
    return newJwt;  
}
```

Claims

Signed with private key

```
public void validateJwt(String jwtParameter, Key key) throws Exception {  
    // Validate the Signed JWT!  
    // Exceptions thrown if not valid  
    Jws<Claims> jwt = Jwts.parser()  
        .setSigningKey(key)  
        .parseClaimsJws(jwtParameter);  
    // Inspect the claims, like make a new JWT  
    // (need a signing key for this)  
    Claims jwtClaims = jwt.getBody();  
    System.out.println(jwtClaims.getAudience(), jwtClaims.getIssuer());  
}
```

Validate against
public key

Signed Requests in Game On

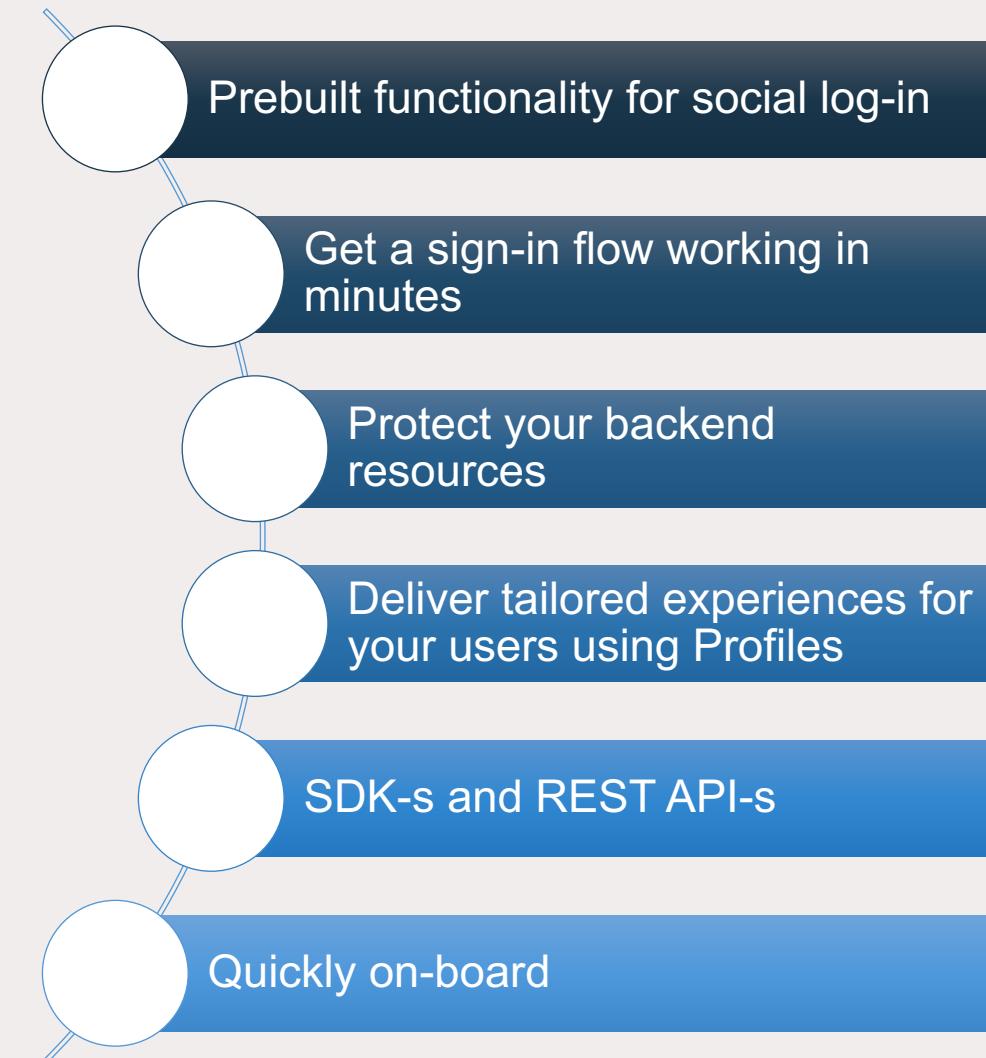
- Service-service operations (internal)
 - Signed with shared secret
- Room handshake for WebSocket
 - Signed with room shared secret
- Room registration
 - Signed with room shared secret
- Uses a shared library - <https://github.com/gameontext/signed>

IBM Bluemix App ID



Helps developers to easily add authentication to their web and mobile apps with few lines of code, and secure their cloud-native applications & services on Bluemix.

App ID also helps manage user specific data that developers can use to build personalized app experiences.



Notices and disclaimers

- Copyright © 2017 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.
- **U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**
- Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed “as is” without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.
- IBM products are manufactured from new parts or new and used parts.
In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply.”
- **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**
- Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.
- References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
- Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.
- It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular, purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli® Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.

InterConnect 2017

