Game On!

Exploring Microservices with a Text-Based Adventure

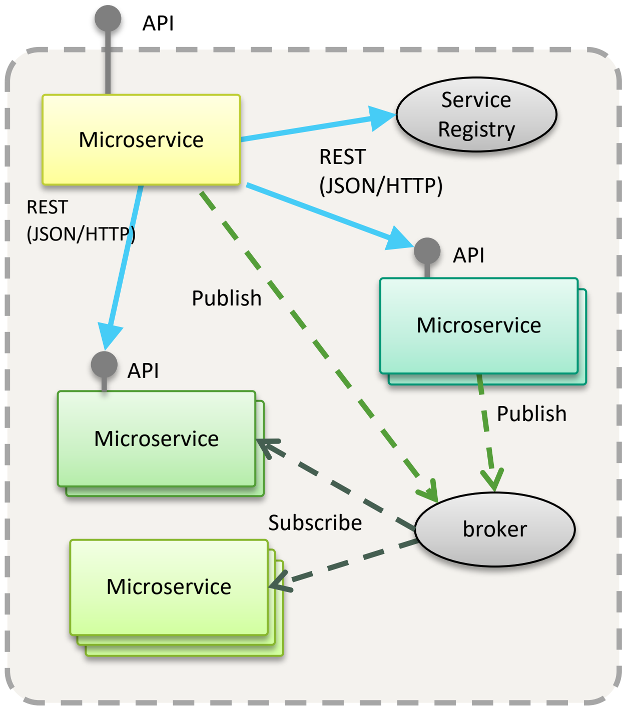
Erin Schnabel @ebullientworks

September 2016

Microservices are used to…

- compose a complex application using
  - "small"
  - independent (autonomous)
  - replaceable
  - processes

- that communicate via
  - language-agnostic APIs

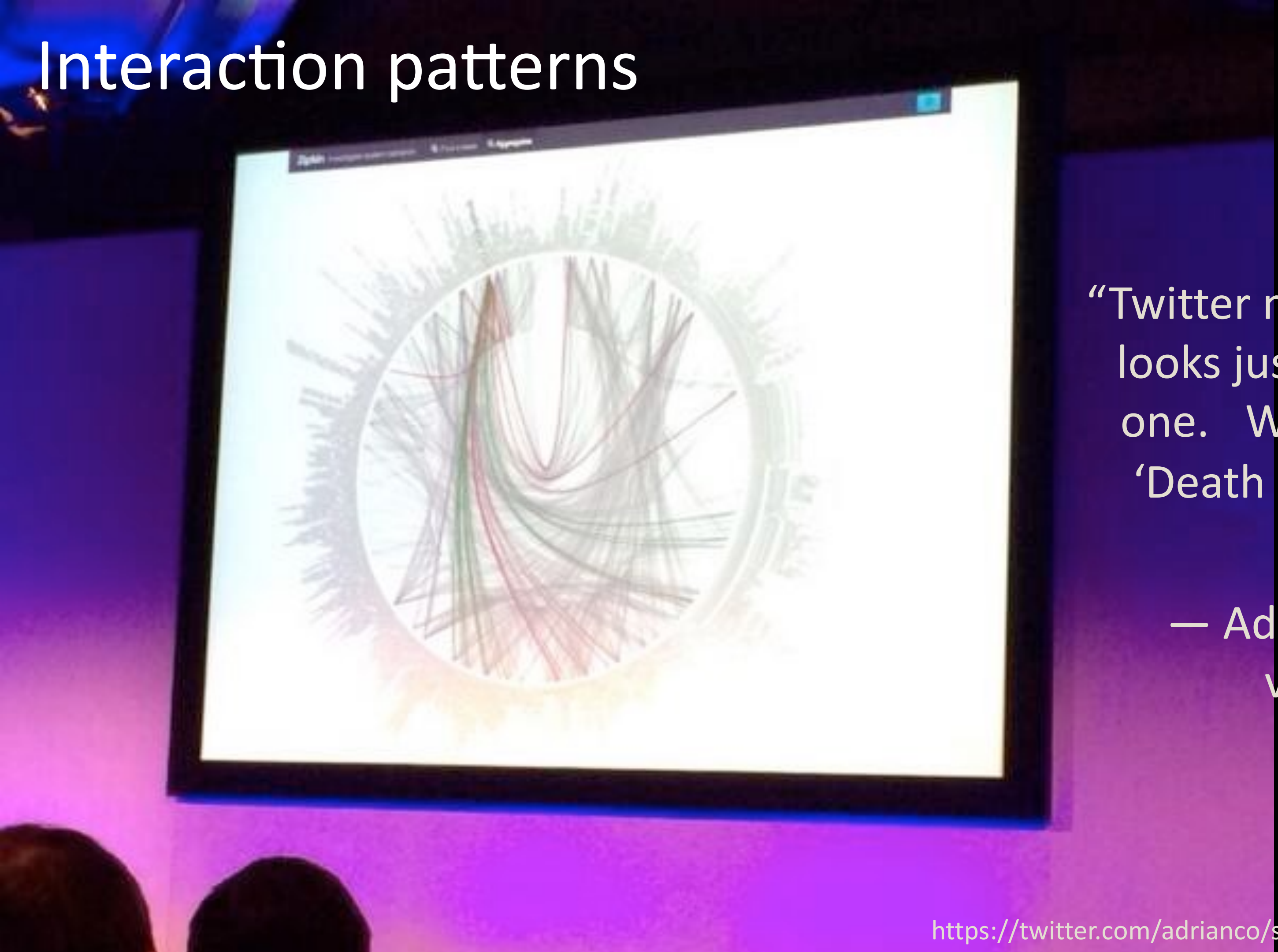


Microservices application

Why?

- Accommodate differences
  - SQL, NoSQL, Graph…
  - Change cycles
  - Scaling profiles
  - Security zoning
- Facilitate growth
  - Polyglot explosion
- Agility!
  - Bounded context (code + data)
  - Faster iteration cycles
- Reduce risk → try new things
  - Isolate legacy vs. unknown

Interaction patterns



"Twitter microservices map looks just like the Netflix one. We called this the 'Death Star' diagram"

— Adrian Cockcroft via twitter

https://twitter.com/adrianco/status/441883572618948608

Fallacies of distributed computing

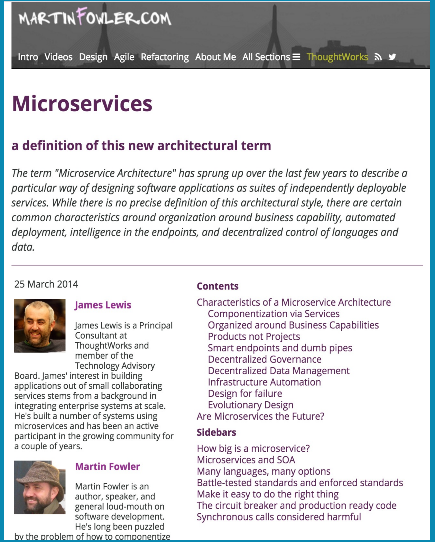
- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn’t change
- There is one administrator
- Transport cost is zero
- The network is homogenous

-- L Peter Deutsch, 1994

https://en.wikipedia.org/wiki/Fallacies_of_distributed_computing

Conway’s law

Bounded Contexts


MARTINFOWLER.COM
Intro Videos Design Agile Refactoring About Me All Sections ThoughtWorks

Microservices

a definition of this new architectural term

The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.

25 March 2014

James Lewis

James Lewis is a Principal Consultant at ThoughtWorks and member of the Technology Advisory Board. James' interest in building applications out of small collaborating services stems from a background in integrating enterprise systems at scale. He's built a number of systems using microservices and has been an active participant in the growing community for a couple of years.

Martin Fowler

Martin Fowler is an author, speaker, and general loud-mouth on software development. He's long been puzzled by the problem of how to componentize

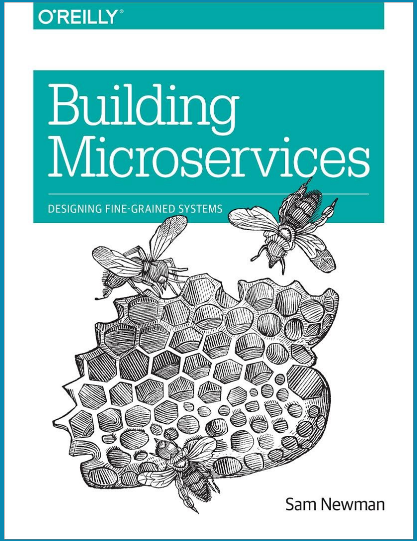
Contents

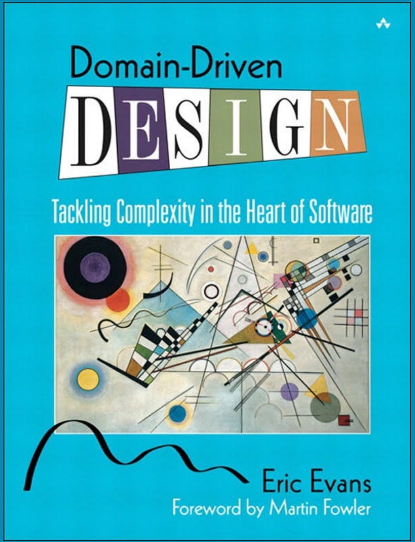
Characteristics of a Microservice Architecture
- Componentization via Services
- Organized around Business Capabilities
- Products not Projects
- Smart endpoints and dumb pipes
- Decentralized Governance
- Decentralized Data Management
- Infrastructure Automation
- Design for failure
- Evolutionary Design

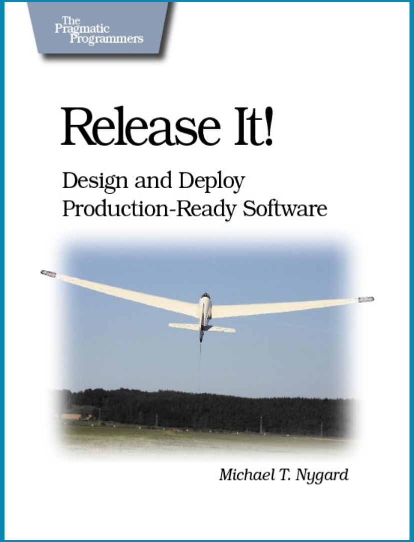
Are Microservices the Future?

Sidebars

How big is a microservice?
Microservices and SOA
Many languages, many options
Battle-tested standards and enforced standards
Make it easy to do the right thing
The circuit breaker and production ready code
Synchronous calls considered harmful




Domain-Driven DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler


The Pragmatic Programmers
Release It!
Design and Deploy Production-Ready Software
Michael T. Nygard

Eventual consistency

DevOps

Testing?

Automation

Microservices Sample Apps…

- Create a si

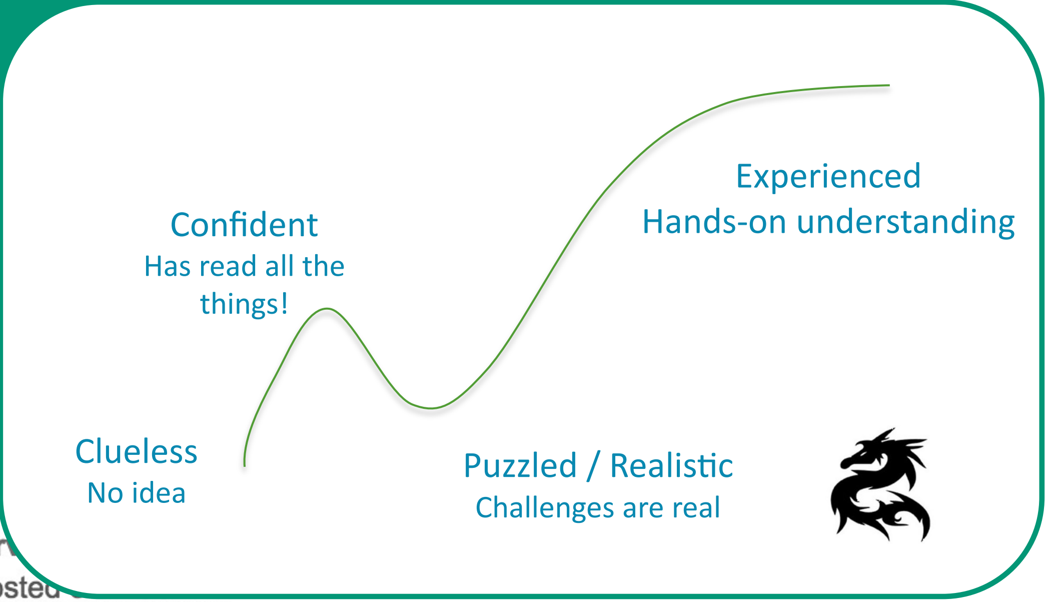
Microservices are so easy!

…vices in 5 minutes | JavaWorld

…ave not imported the sa

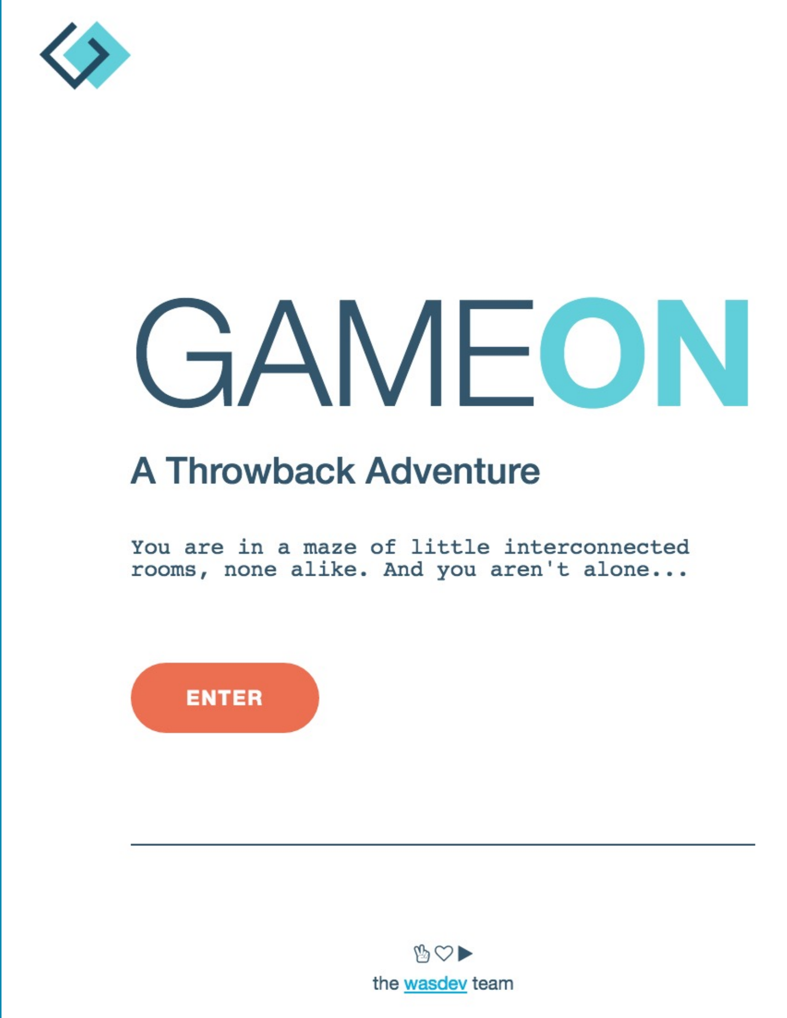
- Rebuild a pre-baked micros

Microservices Onlin
https://developer.ibm.com
Mar 16, 2015 - A **microserv**
Java JAX-RS, PHP and hosted



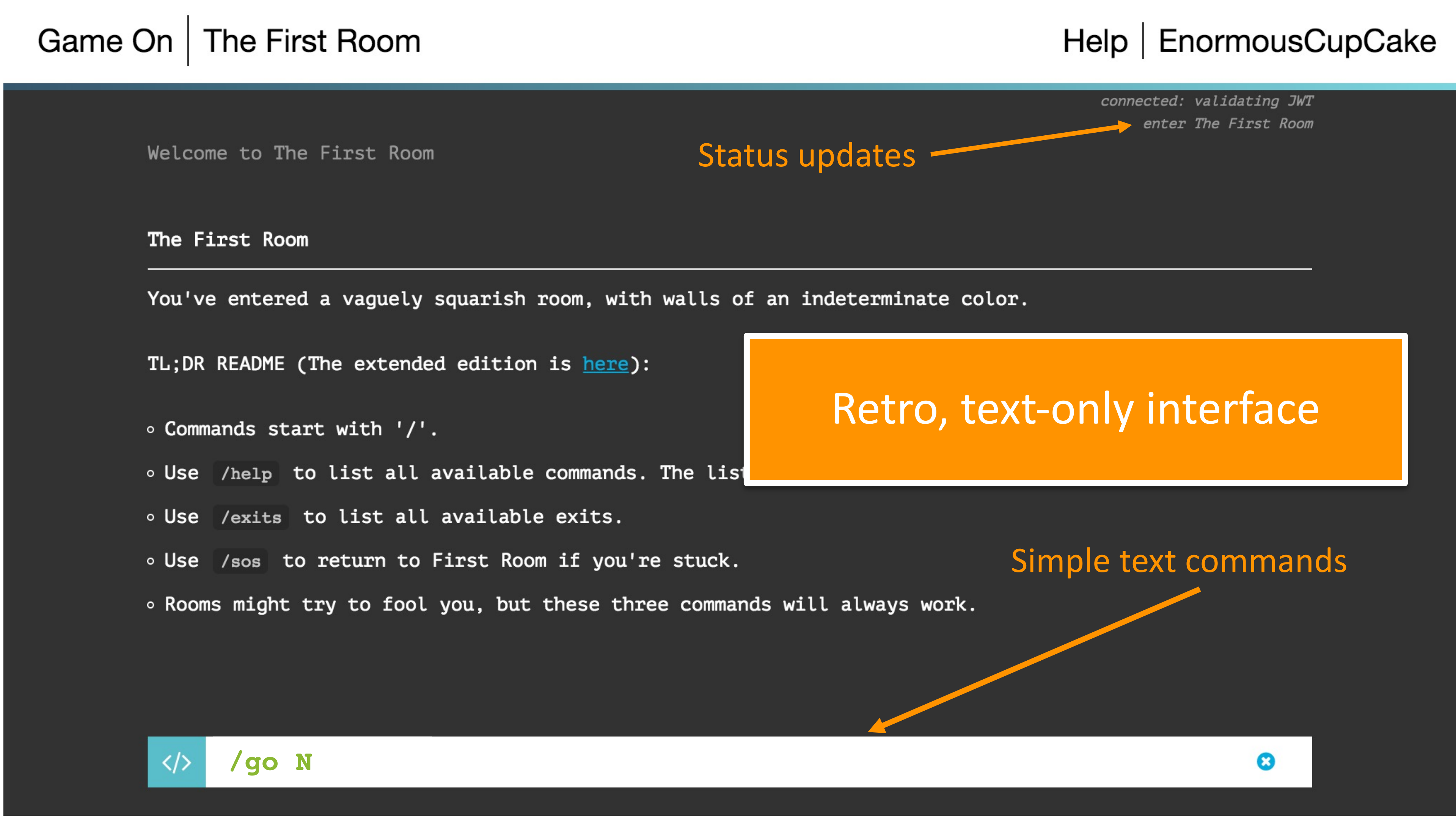
The premise …

- 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...



Game On | The First Room

Help | EnormousCupCake

```
connected: validating JWT
enter The First Room
```

Status updates

```
Welcome to The First Room
```

The First Room

You've entered a vaguely squarish room, with walls of an indeterminate color.

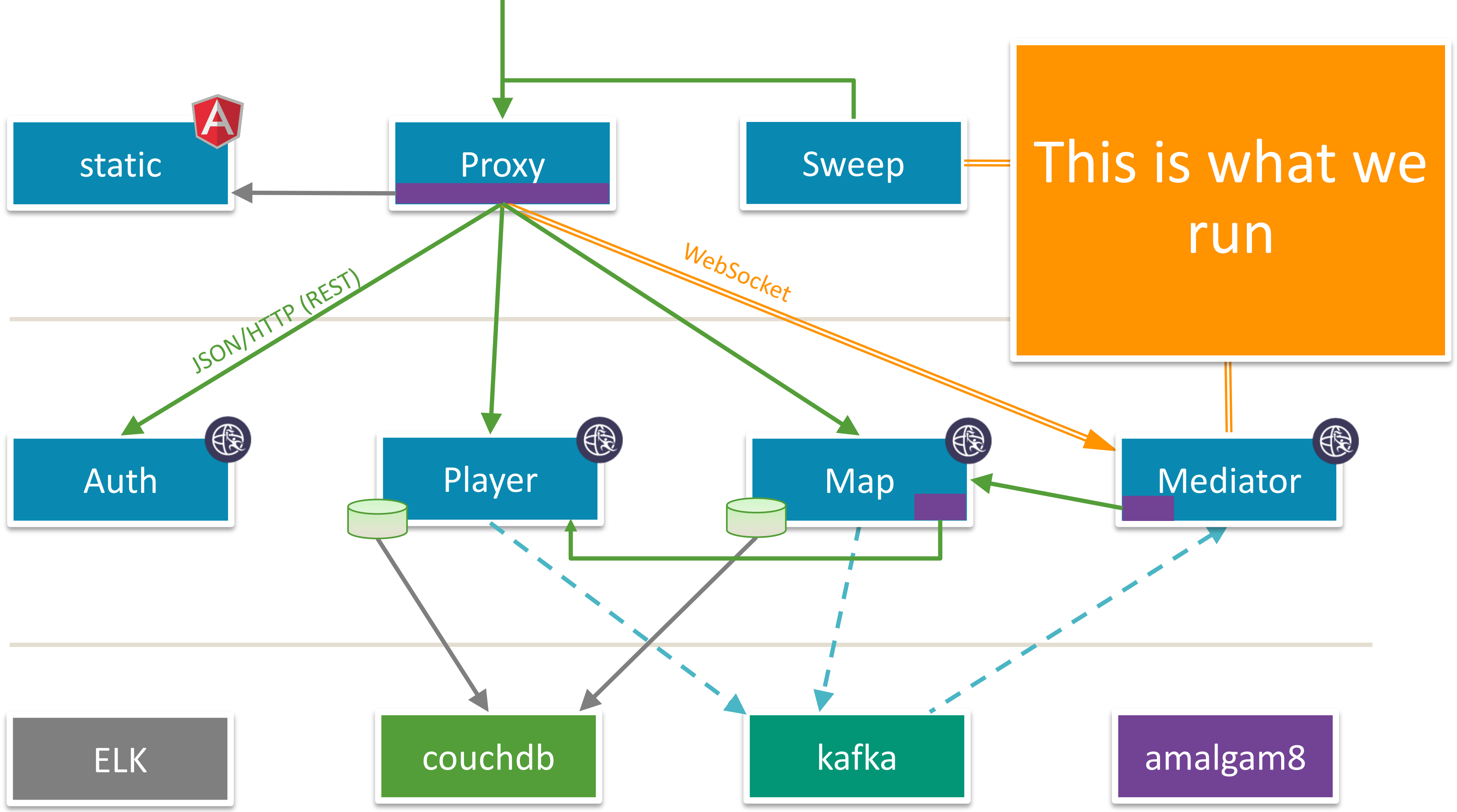
TL;DR README (The extended edition is here):

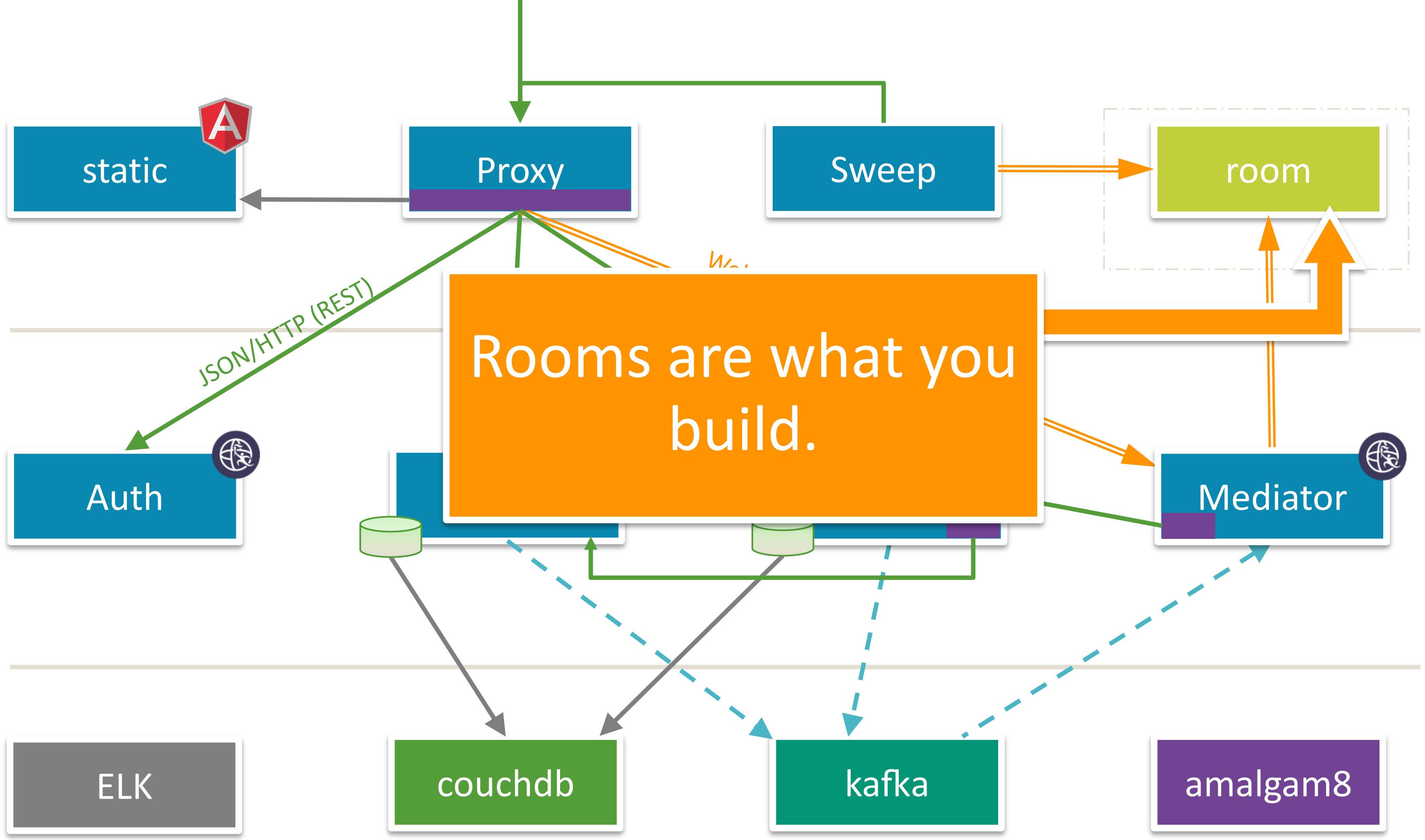
- Commands start with '/'.
- Use `/help` to list all available commands. The list
- Use `/exits` to list all available exits.
- Use `/sos` to return to First Room if you're stuck.
- Rooms might try to fool you, but these three commands will always work.

```
/go N
```

Retro, text-only interface

Simple text commands


static
Proxy
Sweep
This is what we run
JSON/HTTP (REST)
WebSocket
Auth
Player
Map
Mediator
ELK
couchdb
kafka
amalgam8



Rooms are what you build.

What happens when…

1. Build a basic room

2. Scale that room (multiple instances)
   - Where are players?
   - What about items or shared state?
   - Latency, managing calls to additional services

Exploration of solutions for caching, circuit breakers, service interaction patterns

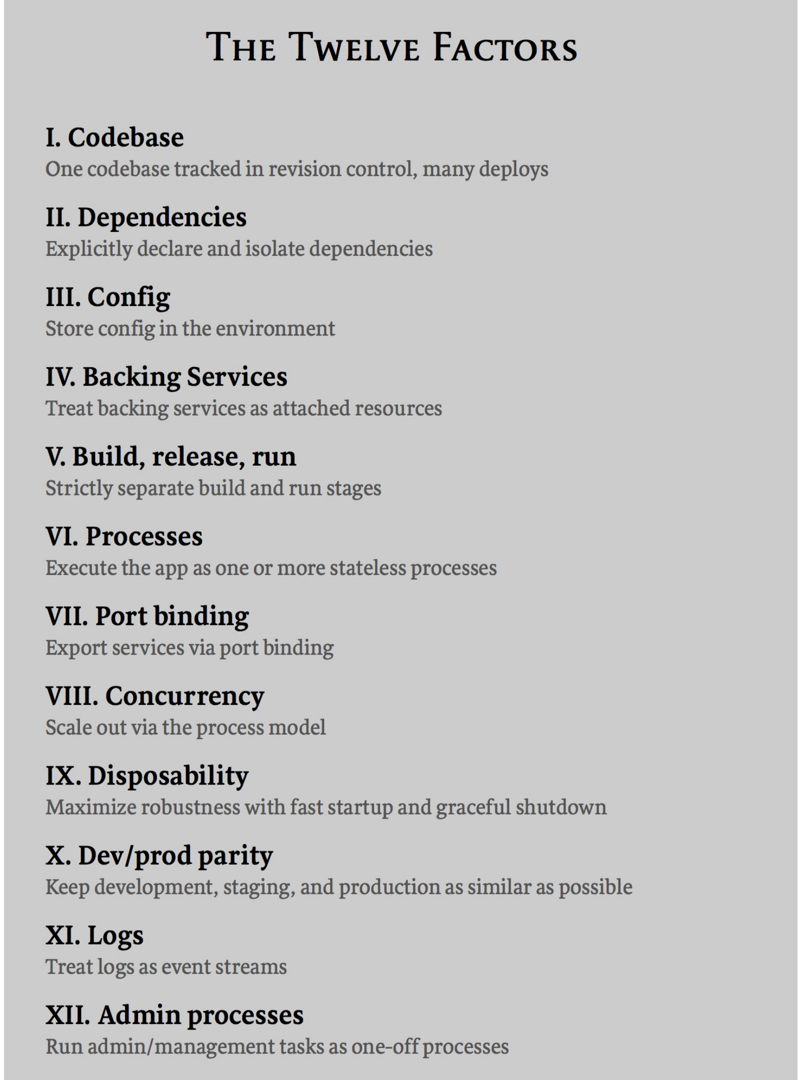
Twelve Factors

Twelve factor applications

- "a methodology for building software-as-a-service applications"
  - Created by developers at Heroku

- Factors are independent of
  - programming language,
  - backing services,
  - cloud provider

- http://12factor.net/



The Twelve Factors

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing Services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

Git + Submodules (Factor 1: codebase)

- Root repository: https://github.com/gameontext/gameon
  - Optional use of submodules
- Key: Only builds update submodule commit levels
  - Prevents conflicts and confusion caused by humans

Containers (Factor 2: dependencies, 5: build/release/run, 6: Processes, 8: concurrency, 10: dev/prod parity)

- Encapsulation of all dependencies
- Parity: dev -> test -> prod
- Configuration passed in via environment

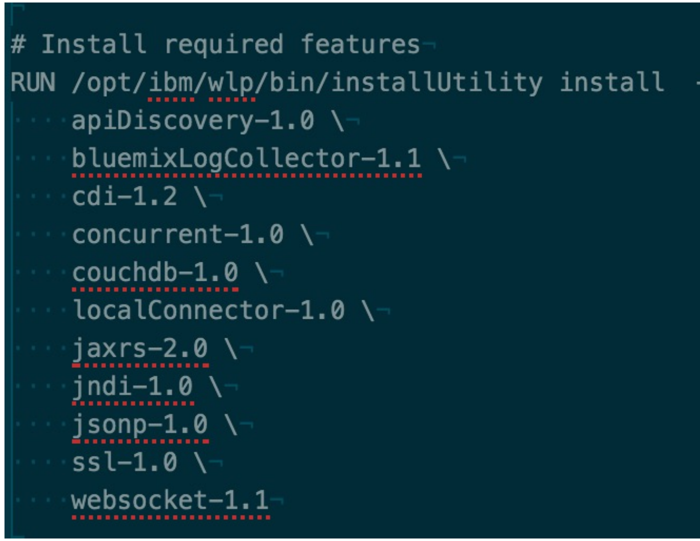
- Local: Docker Compose or Vagrant
  - Pre-built images in dockerhub (this came later..)
  - Overlays for local editing

- Independent build pipelines per service to deploy containers

Liberty (Factor 2, 10, 3: config, 4: backing services, 7: port binding, 9: disposability)

- Java services are Liberty-based
- Customizable features: Cachable Docker Layers
  - Explicit app server dependencies
  - Self-contained immutable artifact
  - Smaller war (smaller delta)

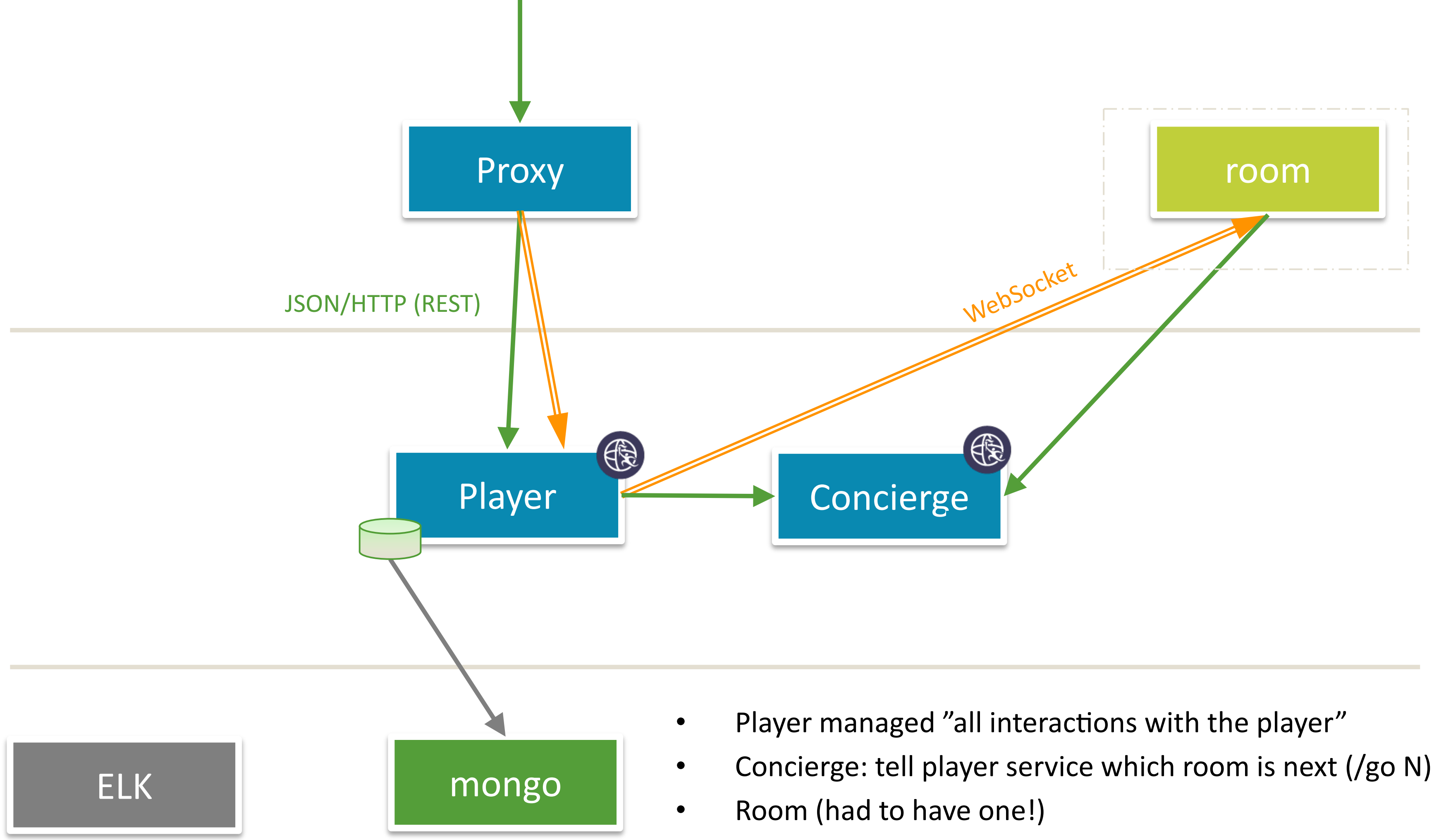
- Environment variables in server config
  - Common configuration across environments
  - Config munging not necessary
  - Composable configuration w/ dropins if required



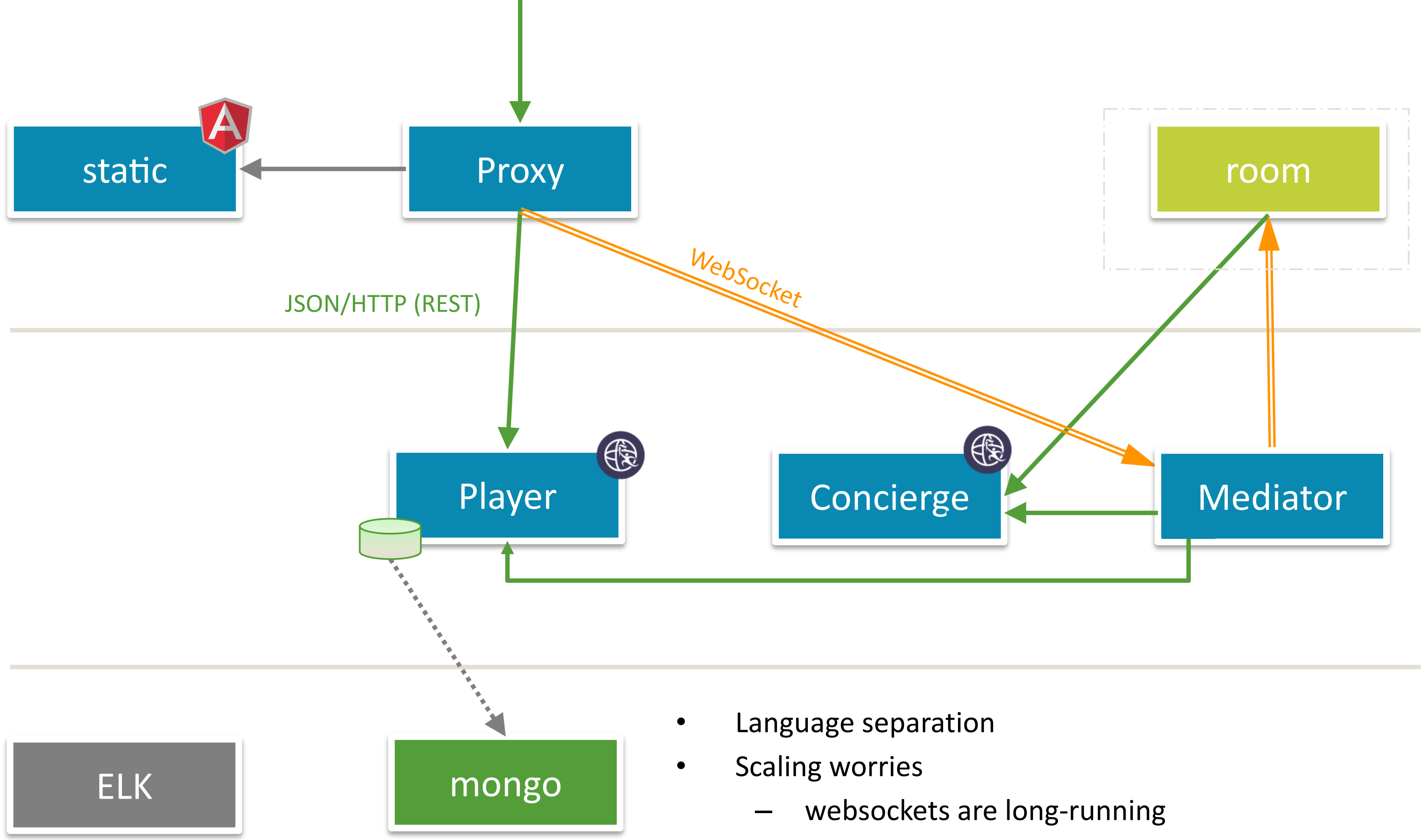
```
# Install required features
RUN /opt/ibm/wlp/bin/installUtility install
    apiDiscovery-1.0 \
    bluemixLogCollector-1.1 \
    cdi-1.2 \
    concurrent-1.0 \
    couchdb-1.0 \
    localConnector-1.0 \
    jaxrs-2.0 \
    jndi-1.0 \
    jsonp-1.0 \
    ssl-1.0 \
    websocket-1.1
```

```
<couchdb id="couchdb"
         jndiName="couchdb/connector"
         libraryRef="couchdb-lib"
         password="${env.COUCHDB_PASSWORD}"
         url="${env.COUCHDB_SERVICE_URL}"
         username="${env.COUCHDB_USER}"/>
```

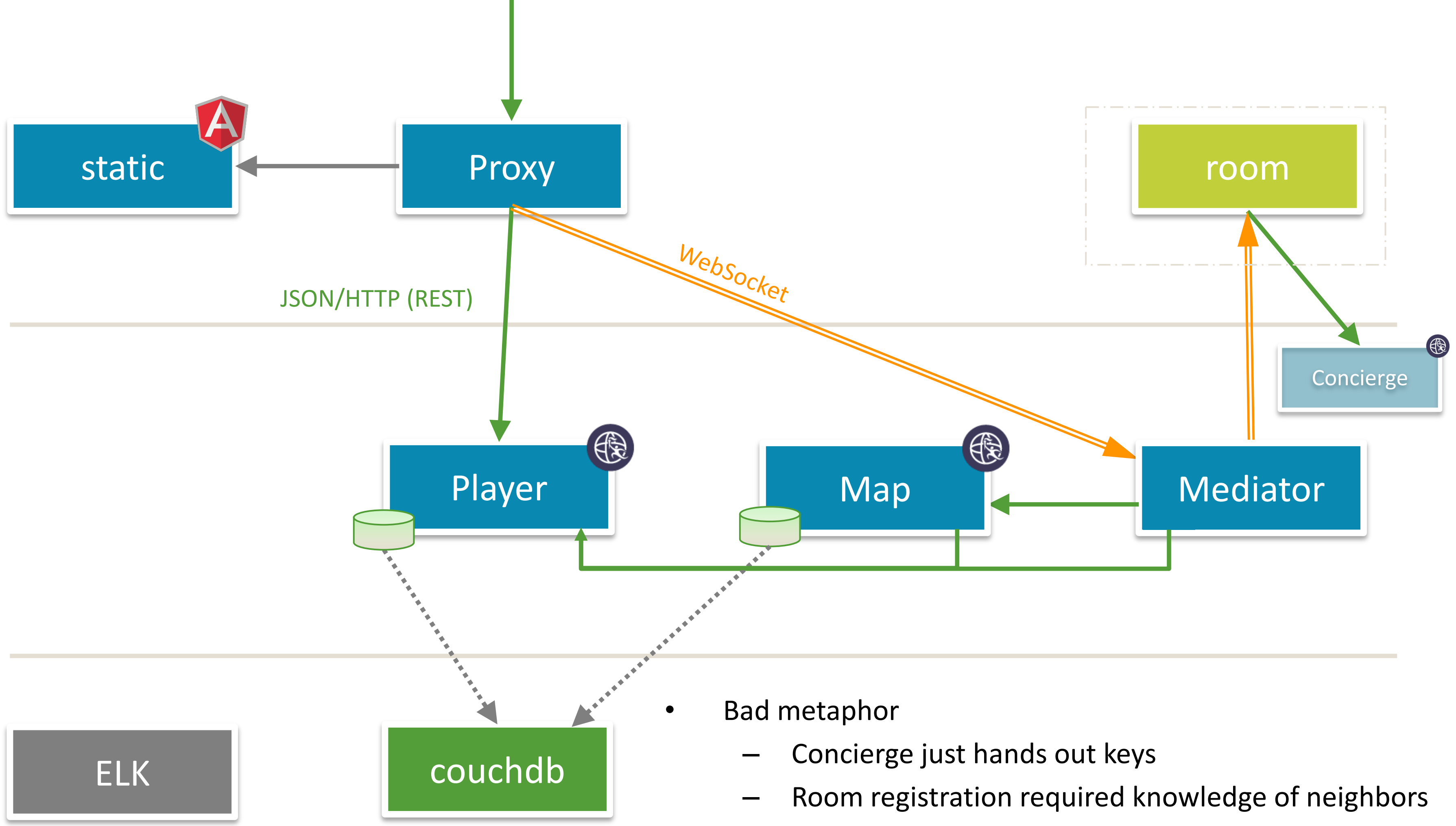
Service composition



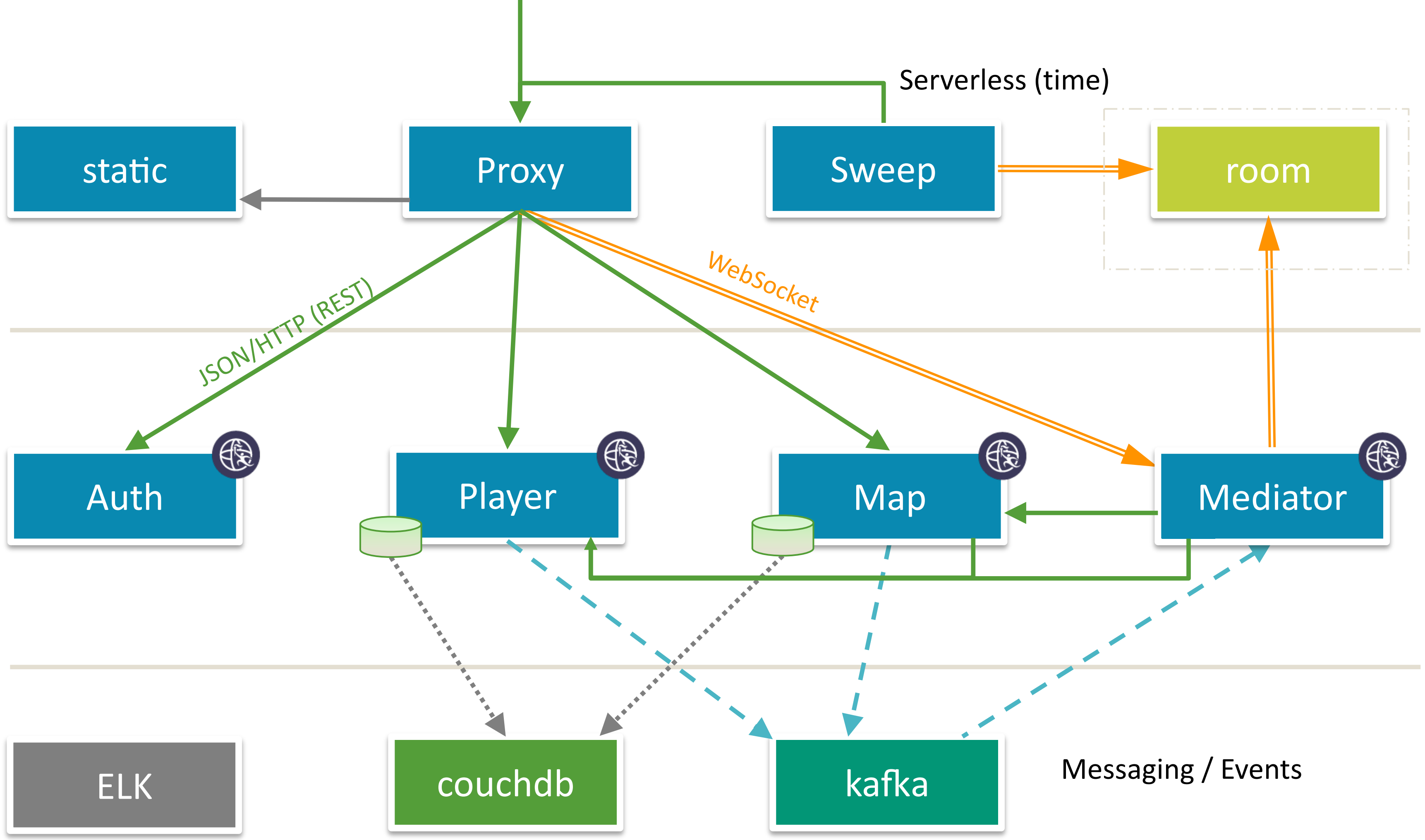
- Player managed ”all interactions with the player”
- Concierge: tell player service which room is next (/go N)
- Room (had to have one!)



- Language separation
- Scaling worries
  - websockets are long-running
  - Simple CRUD operations



- Bad metaphor
  - Concierge just hands out keys
  - Room registration required knowledge of neighbors

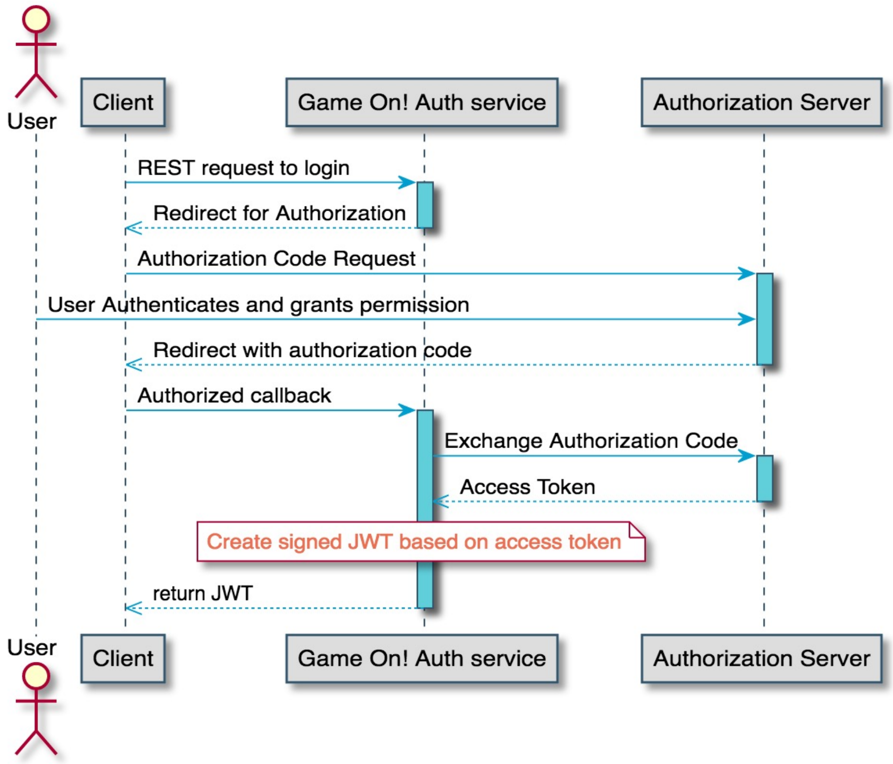

Serverless (time)
static
Proxy
Sweep
room
JSON/HTTP (REST)
WebSocket
Auth
Player
Map
Mediator
ELK
couchdb
kafka
Messaging / Events

Security

OAuth & JWTs

- OAuth proxy
  - Application id w/ different front-end
  - Could be a gateway instead

- Access token converted into signed JWT
- System services deal only with JWT
  - game-on.org SSL certificate
  - Well-known public key


User
Client
Game On! Auth service
Authorization Server
REST request to login
Redirect for Authorization
Authorization Code Request
User Authenticates and grants permission
Redirect with authorization code
Authorized callback
Exchange Authorization Code
Access Token
Create signed JWT based on access token
return JWT
User
Client
Game On! Auth service
Authorization Server

Hashed message authentication codes (HMACs)

- Shared secrets
  - Credentials not sent on the wire
  - Used to verify identity of sender

- Map operations
  - Mutable operations require HMAC signature
  - Hashed signature used to prevent replays

- Room handshake for WebSocket
  - It is the game calling the room
  - Room answering the game

- https://book.game-on.org/microservices/ApplicationSecurity.html

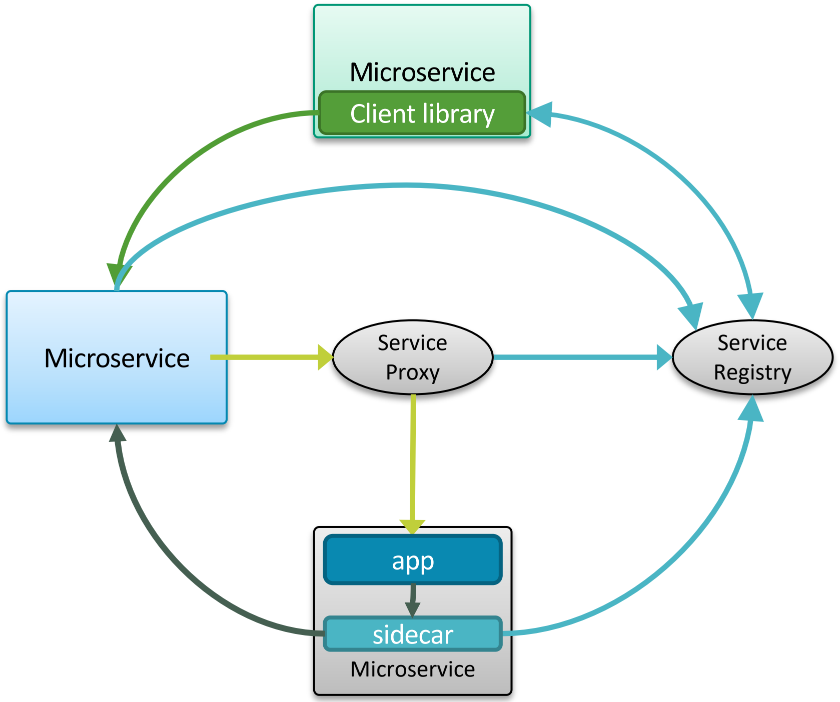
Shared Library

Service discovery

Service registration and discovery

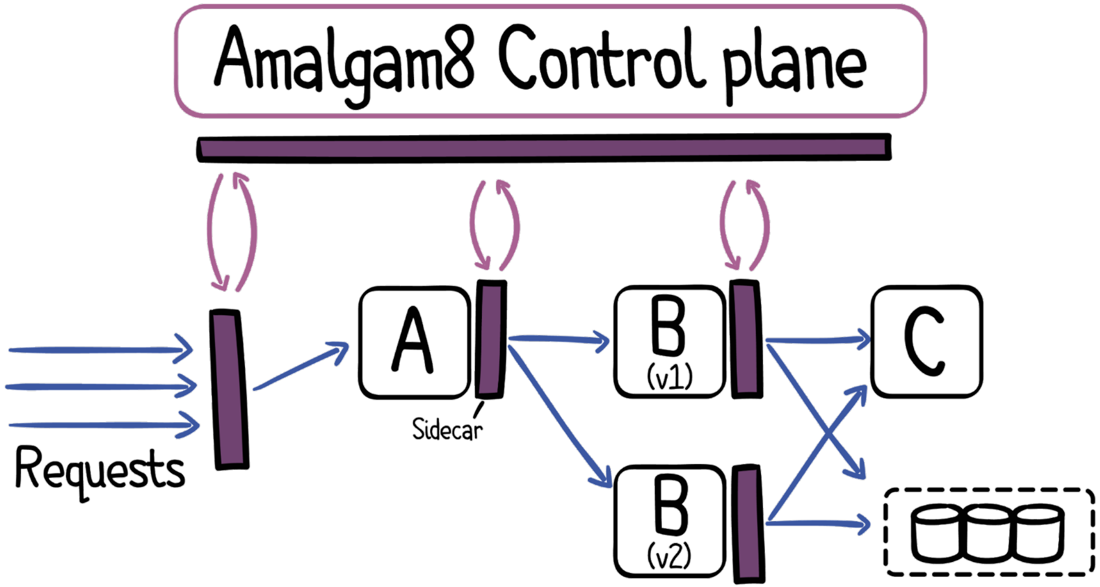
- Required for load balancing and scaling
- Services need to find each other
- Environment changes constantly

- Client-side or server-side?
- Client library, sidecar, or proxy?



Amalgam8

- Basics
  - Service registration
  - Service discovery
  - Client-side load balancing



- How
  - Sidecar model. 2 options:
    - An independent process running in the same container as the app
    - An independent container running in the same pod as the app container

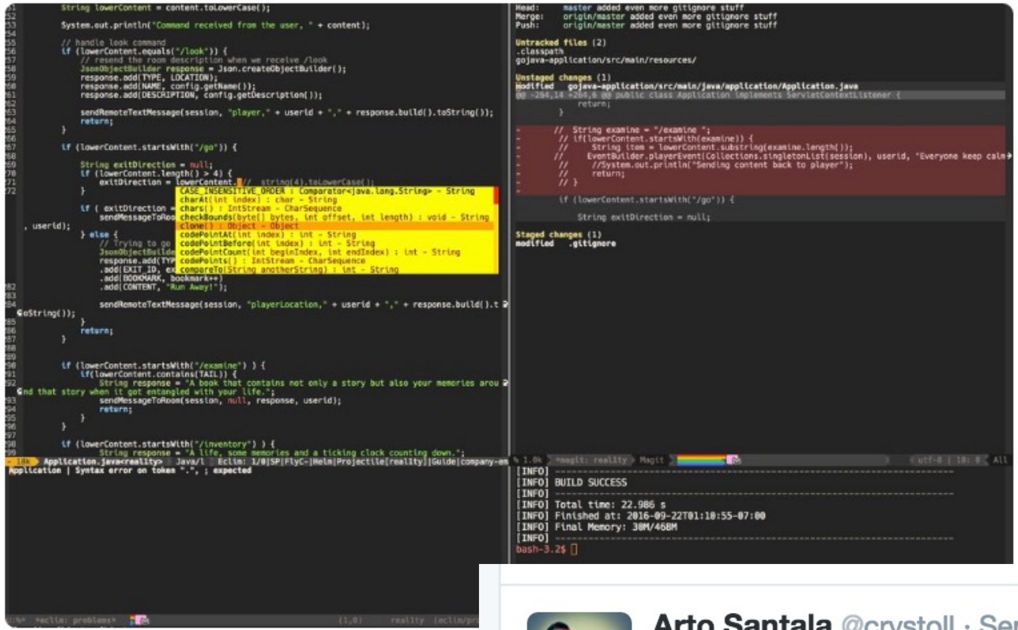
Successful?

swardley @swardley · Aug 2
One I want to try -> Learning microservices in the open with GameOn! by **@ebullientworks** conferences.oreilly.com/oscon/open-sou… #oscon

Learning microservices in the open with GameOn! -...
There are plenty of talks out there about how to get started with microservices, but in reality you learn by doing. Erin Schnabel and Katherine Stanley explore l...
conferences.oreilly.com

♥ 2 •••

ha! got it working in emacs so I dont have to much around in those ides that drive me up the wall! Now I can go to sleep.



```
        String lowerContent = content.toLowerCase();

        System.out.println("Command received from the user, " + content);

        // handle look command
        if (lowerContent.equals("/look")) {
            // resend the room description when we receive /look
            JsonObjectBuilder response = Json.createObjectBuilder();
            response.add(TYPE, LOCATION);
            response.add(NAME, config.getName());
            response.add(DESCRIPTION, config.getDescription());

            sendRemoteTextMessage(session, "player," + userid + "," + response.build().toString());
            return;
        }

        if (lowerContent.startsWith("/go")) {

            String exitDirection = null;
            if (lowerContent.length() > 4) {
                exitDirection = lowerContent.//  string(4).toLowerCase();
            }                   CASE_INSENSITIVE_ORDER : Comparator<java.lang.String> - String
                                charAt(int index) : char - String
            if ( exitDirection  chars() : IntStream - CharSequence
                sendMessageToRoo checkBounds(byte[] bytes, int offset, int length) : void - String
, userid);                      clone() : Object - Object
            } else {            codePointAt(int index) : int - String
                // Trying to go codePointBefore(int index) : int - String
                JsonObjectBuilde codePointCount(int beginIndex, int endIndex) : int - String
                response.add(TYP codePoints() : IntStream - CharSequence
                .add(EXIT_ID, ex compareTo(String anotherString) : int - String
                .add(BOOKMARK, bookmark++)
                .add(CONTENT, "Run Away!");

                sendRemoteTextMessage(session, "playerLocation," + userid + "," + response.build().t
oString());
            }
            return;
        }


        if (lowerContent.startsWith("/examine") ) {
            if(lowerContent.contains(TAIL)) {
                String response = "A book that contains not only a story but also your memories arou
nd that story when it got entangled with your life.";
                sendMessageToRoom(session, null, response, userid);
                return;
            }
        }

        if (lowerContent.startsWith("/inventory") ) {
                String response = "A life, some memories and a ticking clock counting down.";
- 18k  Application.java<reality>  Java/l  Eclim: 1/0|SP|FlyC-|Helm|Projectile[reality]|Guide|company-em
Application | Syntax error on token ".", ; expected
```

```
Head:     master added even more gitignore stuff
Merge:    origin/master added even more gitignore stuff
Push:     origin/master added even more gitignore stuff

Untracked files (2)
.classpath
gojava-application/src/main/resources/

Unstaged changes (1)
modified   gojava-application/src/main/java/application/Application.java
@@ -264,14 +264,6 @@ public class Application implements ServletContextListener {
             return;
         }

-        //  String examine = "/examine ";
-         // if(lowerContent.startsWith(examine)) {
-         //     String item = lowerContent.substring(examine.length());
-        //     EventBuilder.playerEvent(Collections.singletonList(session), userid, "Everyone keep calm
-         //     //System.out.println("Sending content back to player");
-         //     return;
-         // }
-
         if (lowerContent.startsWith("/go")) {

             String exitDirection = null;

Staged changes (1)
modified   .gitignore
```

```
% 1.0k  *magit: reality  Magit                              utf-8 | 10: 0  All
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 22.986 s
[INFO] Finished at: 2016-09-22T01:10:55-07:00
[INFO] Final Memory: 30M/468M
[INFO] ------------------------------------------------------------------------
bash-3.2$
```

4h

Arto Santala @crystoll · Sep 19
@ebullientworks @Dev_Events @gameontext Cool, looking forward to try it!

1 ♥ 1 •••

Questions?

Thank You!

Play – http://game-on.org

Learn more – http://book.game-on.org

Erin Schnabel | schnabel@us.ibm.com | @ebullientworks