INDEX

SAN FRANCISCO

Discover. Collaborate. Deploy.

Building Cloud Native Applications: Best Practices in Action

Erin Schnabel

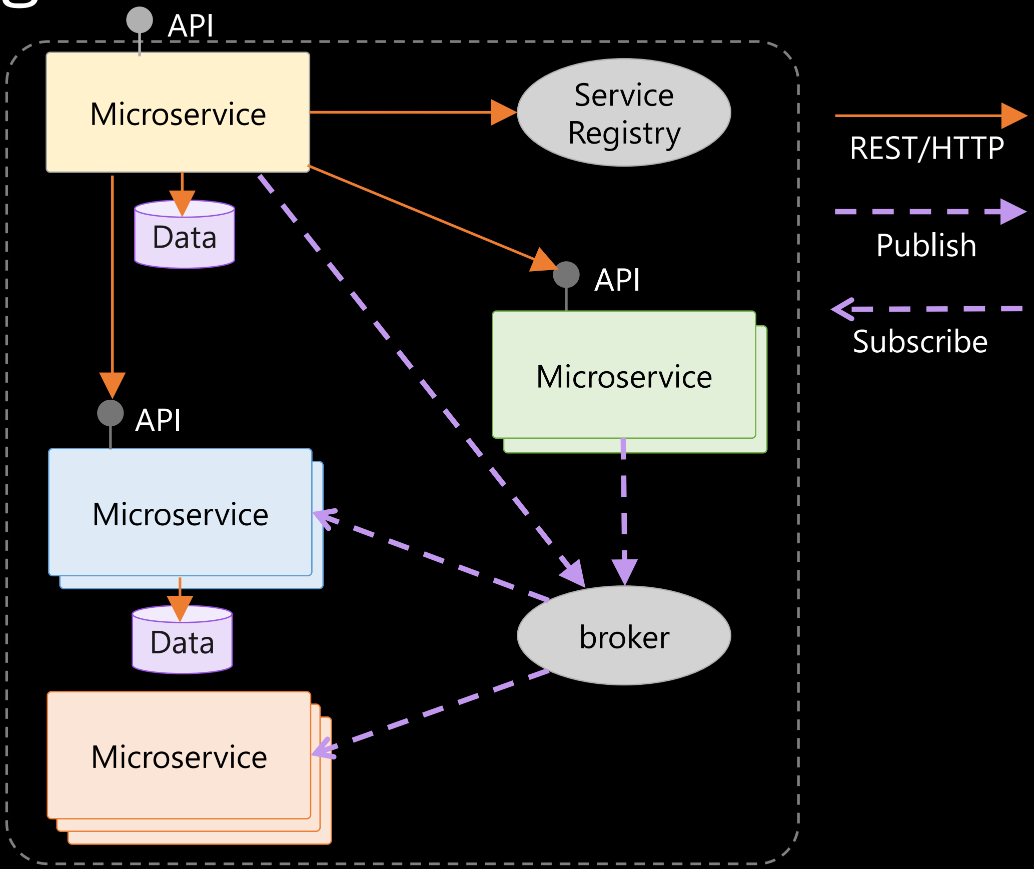
@ebullientworks

Cloud Native

An **application architecture** designed to leverage the **strengths** and accommodate the **challenges** of a **standardized** cloud environment, including concepts such as **elastic** scaling, **immutable** deployment, **disposable** instances, and **less predictable infrastructure**.

Microservices are used to...

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

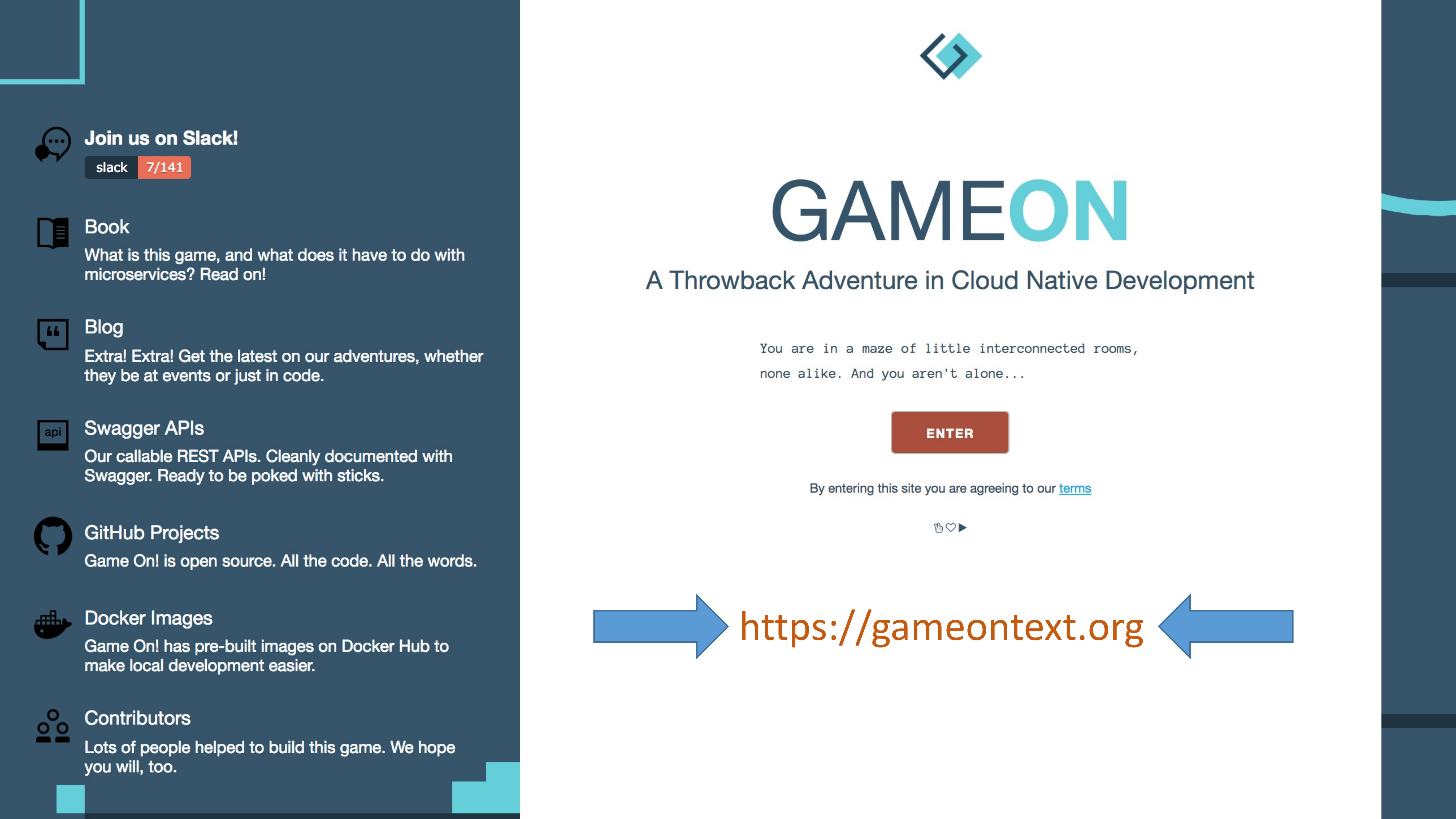


All the words ..

This workshop will review the **best practices in building Microservices** using **popular cloud-native programming models**. It will address questions such as:

- **Where do we start?** How big is a Microservice?
- Should every Microservice **own its own data**?
- How much … belongs in **the application vs. the environment**?
- How can we leverage … **Kubernetes and Istio**?

In this workshop we will address the above questions, outline some best practices and share our **experience gained building Game On!**, an open source Microservices application built to help people explore cloud-native environments.



Join us on Slack!

slack 7/141

Book

What is this game, and what does it have to do with microservices? Read on!

Blog

Extra! Extra! Get the latest on our adventures, whether they be at events or just in code.

Swagger APIs

Our callable REST APIs. Cleanly documented with Swagger. Ready to be poked with sticks.

GitHub Projects

Game On! is open source. All the code. All the words.

Docker Images

Game On! has pre-built images on Docker Hub to make local development easier.

Contributors

Lots of people helped to build this game. We hope you will, too.

GAMEON

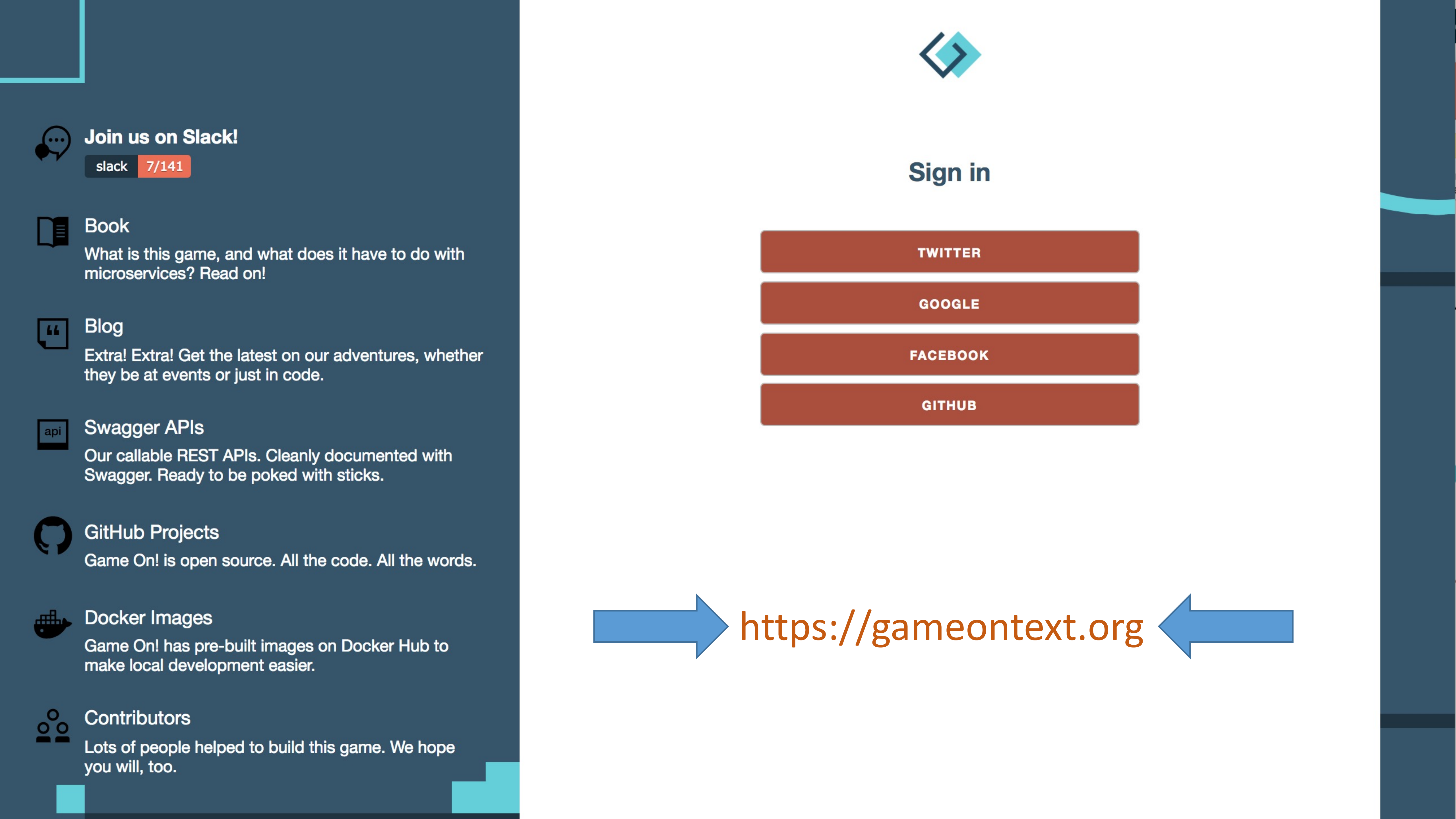
A Throwback Adventure in Cloud Native Development

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

ENTER

By entering this site you are agreeing to our terms

https://gameontext.org



Join us on Slack!

Book

What is this game, and what does it have to do with microservices? Read on!

Blog

Extra! Extra! Get the latest on our adventures, whether they be at events or just in code.

Swagger APIs

Our callable REST APIs. Cleanly documented with Swagger. Ready to be poked with sticks.

GitHub Projects

Game On! is open source. All the code. All the words.

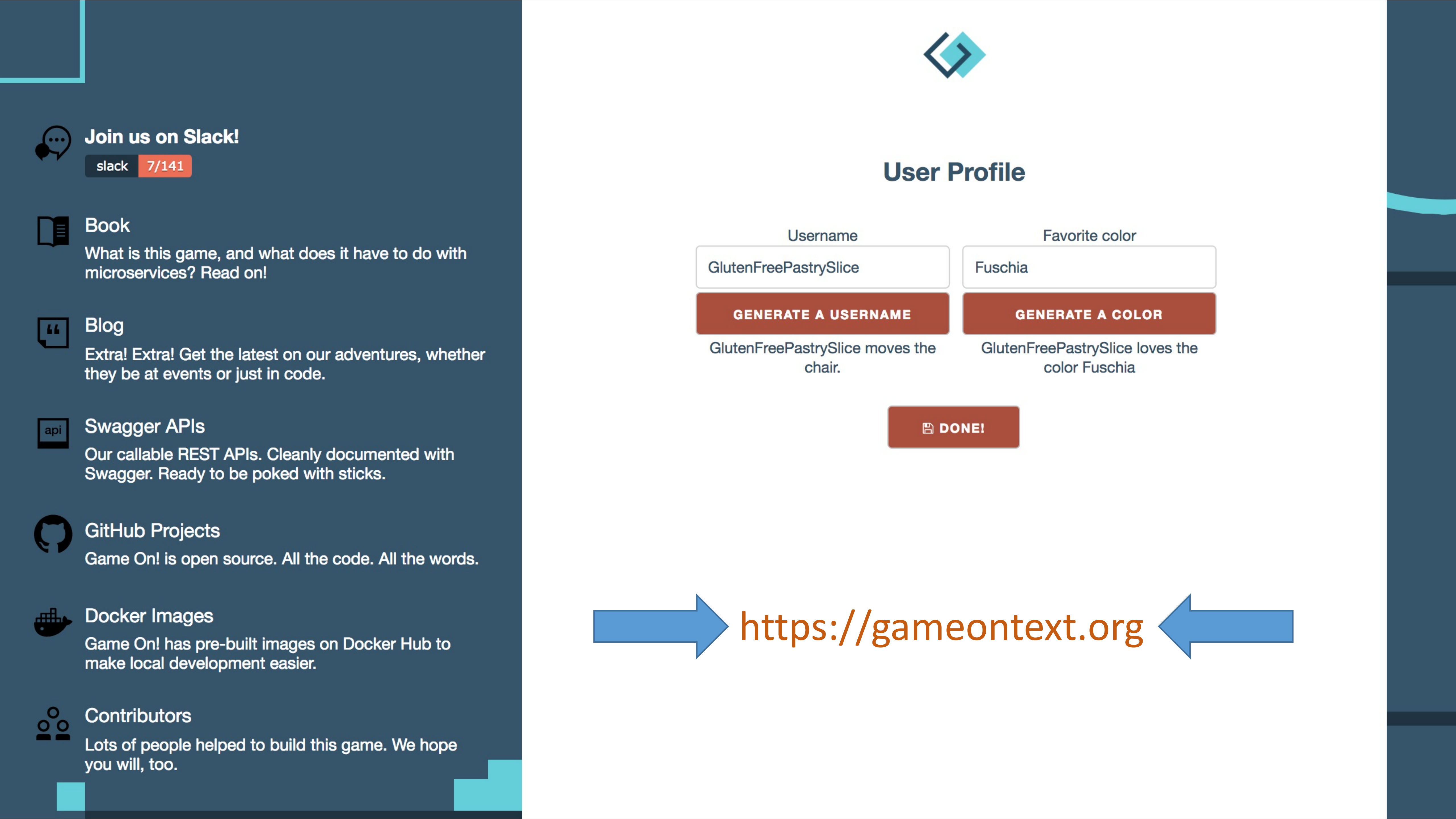
Docker Images

Game On! has pre-built images on Docker Hub to make local development easier.

Contributors

Lots of people helped to build this game. We hope you will, too.

https://gameontext.org



Join us on Slack!

slack 7/141

Book

What is this game, and what does it have to do with microservices? Read on!

Blog

Extra! Extra! Get the latest on our adventures, whether they be at events or just in code.

Swagger APIs

Our callable REST APIs. Cleanly documented with Swagger. Ready to be poked with sticks.

GitHub Projects

Game On! is open source. All the code. All the words.

Docker Images

Game On! has pre-built images on Docker Hub to make local development easier.

Contributors

Lots of people helped to build this game. We hope you will, too.

User Profile

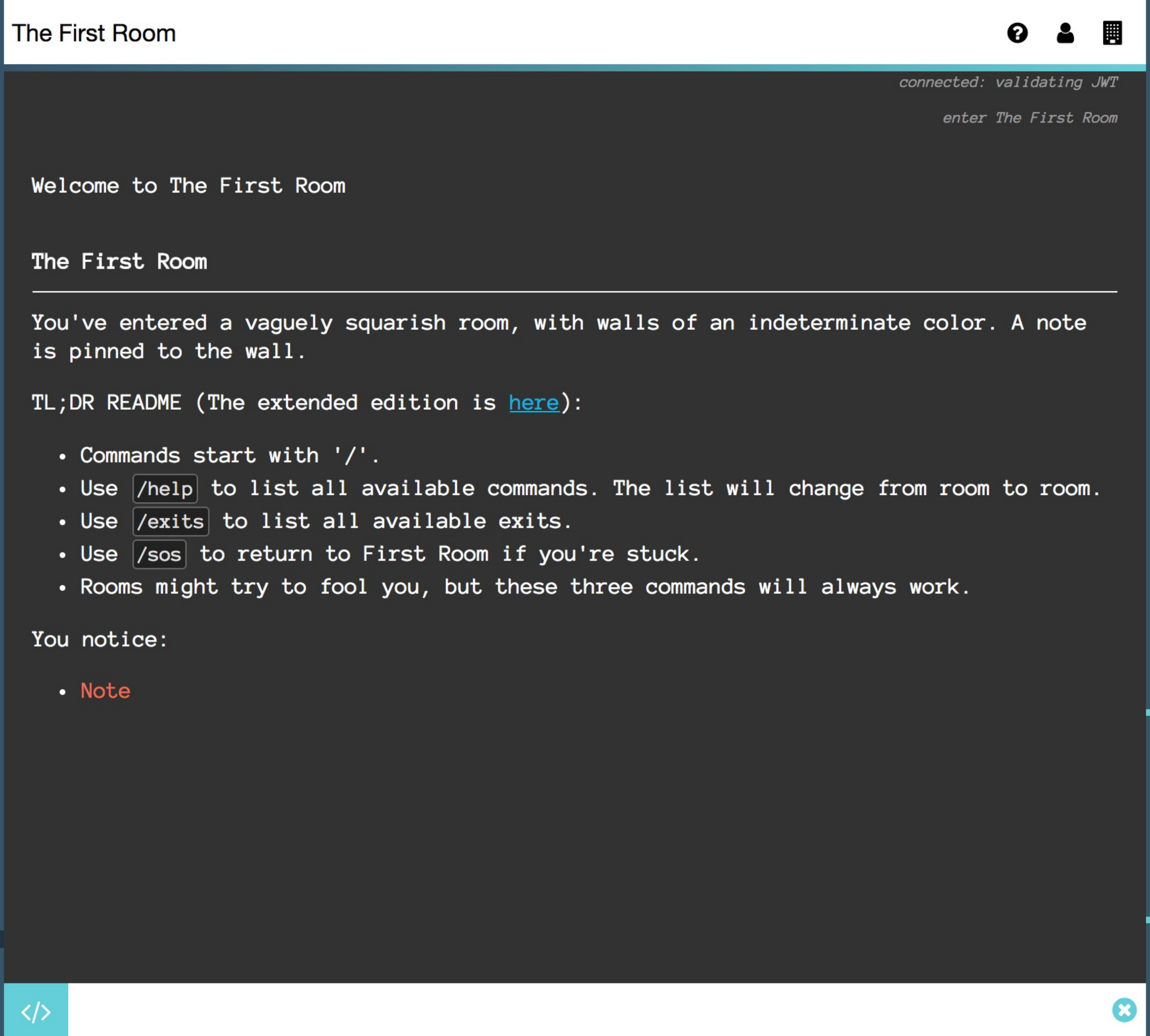
Username: GlutenFreePastrySlice

GlutenFreePastrySlice moves the chair.

Favorite color: Fuschia

GlutenFreePastrySlice loves the color Fuschia

https://gameontext.org



The First Room

connected: validating JWT

enter The First Room

Welcome to The First Room

The First Room

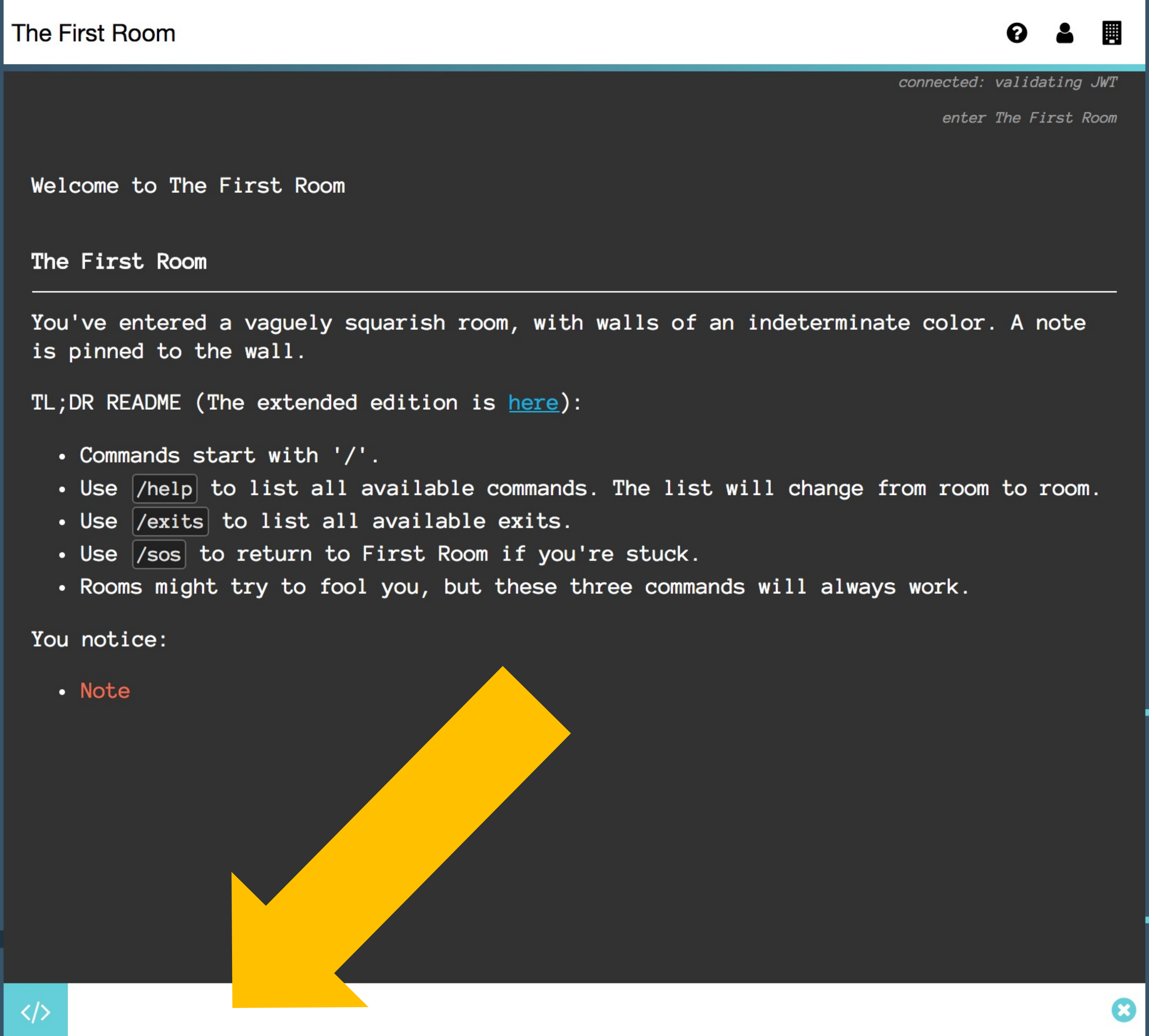
You've entered a vaguely squarish room, with walls of an indeterminate color. A note is pinned to the wall.

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

- Commands start with '/'.
- Use `/help` to list all available commands. The list will change from room to room.
- 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.

You notice:

- Note



The First Room

connected: validating JWT

enter The First Room

Welcome to The First Room

The First Room

You've entered a vaguely squarish room, with walls of an indeterminate color. A note is pinned to the wall.

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

- Commands start with '/'.
- Use `/help` to list all available commands. The list will change from room to room.
- 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.

You notice:

- Note

Things to try

- /go <direction: E|W|N|S>
- /help – Rooms provide additional/custom commands
- /sos – Emergency return to First Room

- From First Room:
  - /go W to ”Junky Place” – written by an 8 year old
  - /go E to “Rec Room” – this room is a puzzle

Pause and think …

- How would you build this?

- **Where do we start?**
- **How big** is a Microservice?
- Should every Microservice **own its own data**?

2 minutes … GO!

Let’s start some background tasks

- Start the path for Local room development:
  - https://github.com/gameontext/gameon/

- Choose your own adventure:
  1. Docker Compose ←
  2. *Kubernetes*
     - *Kubernetes + Helm*

TL;DR

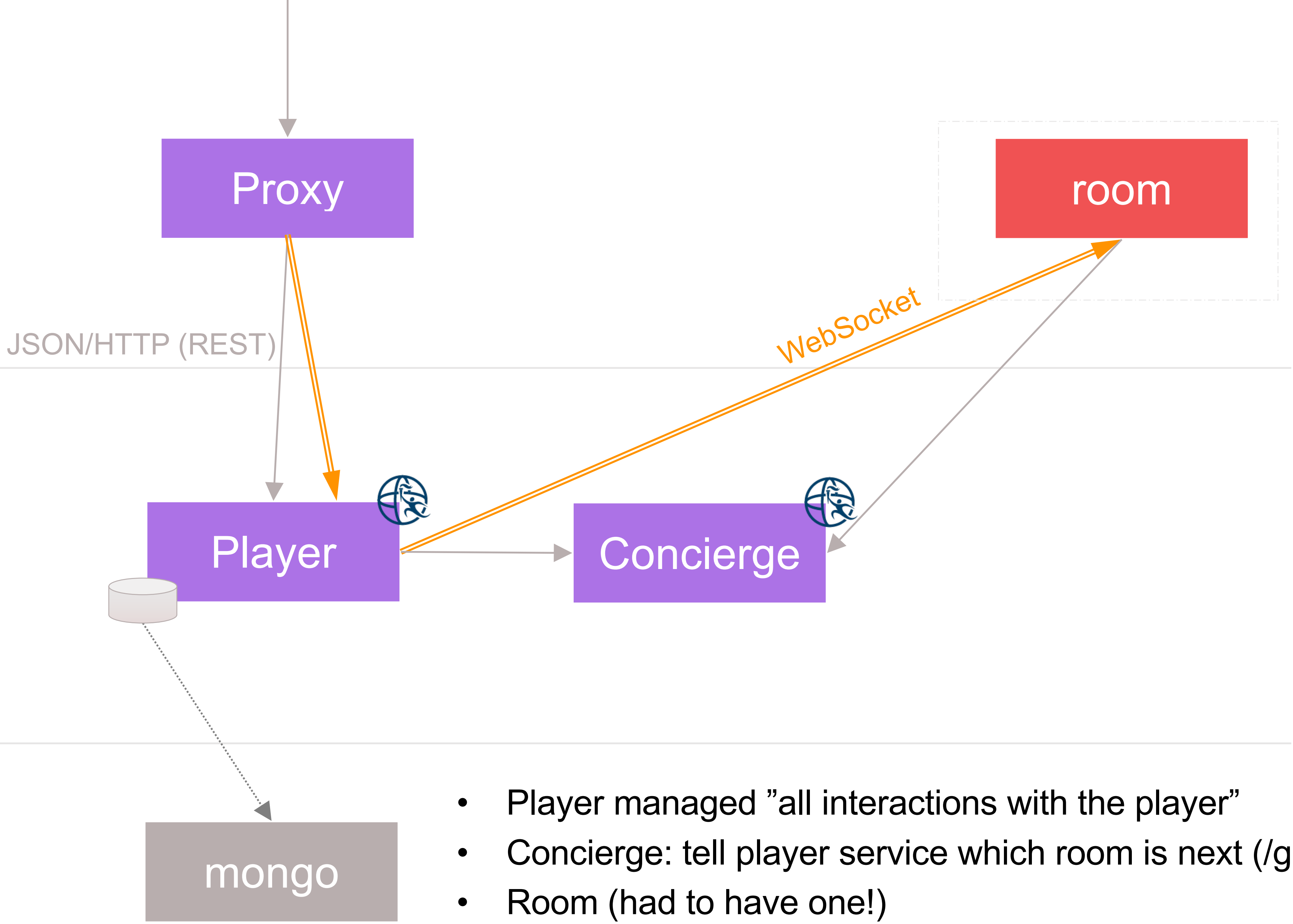
```
$ git clone https://github.com/gameontext/gameon.git

$ cd gameon                     # cd into the project directory

$ ./go-admin.sh choose          # choose Docker Compose (1)

$ eval $(./go-admin.sh env)     # set aliases for admin scripts

$ alias go-run                  # confirm path (docker)

$ go-run setup

$ go-run up

$ go-run wait                  # make sure things are good to go
```

Pause and think ...

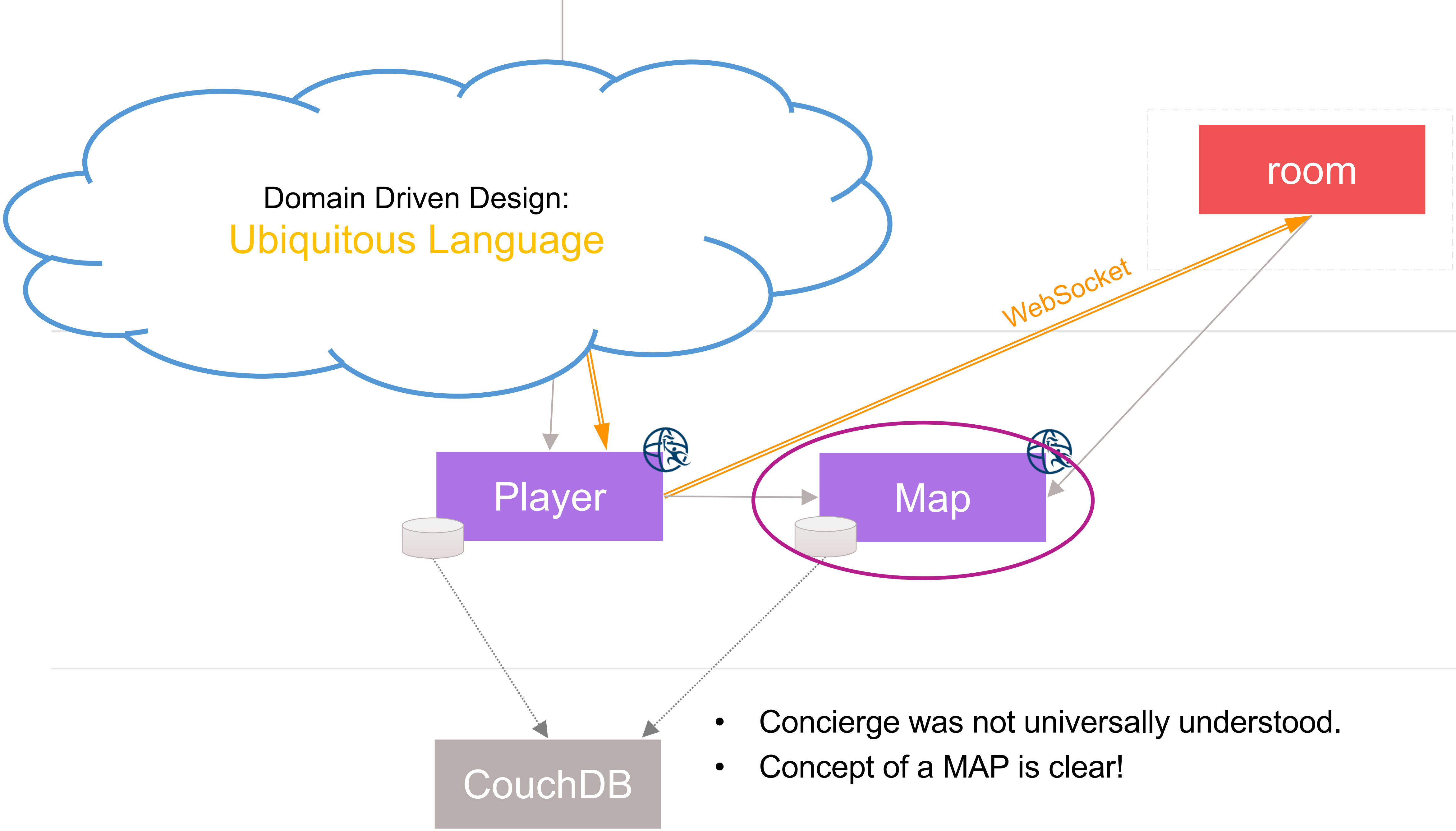
- How would you build this?

How many services did you guess?

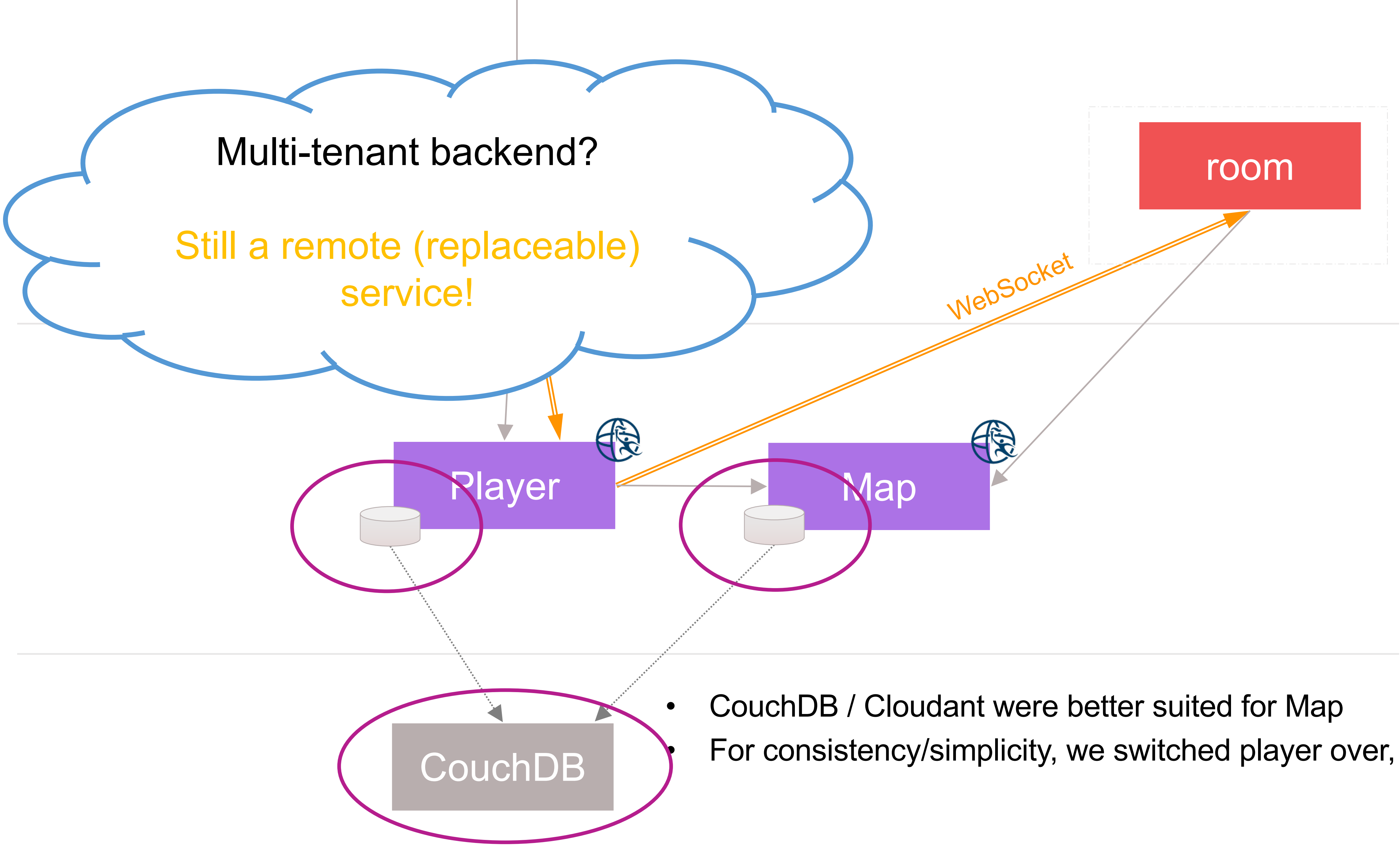
First pass



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



- Concierge was not universally understood.
- Concept of a MAP is clear!



Multi-tenant backend?

Still a remote (replaceable) service!

room

WebSocket

Player

Map

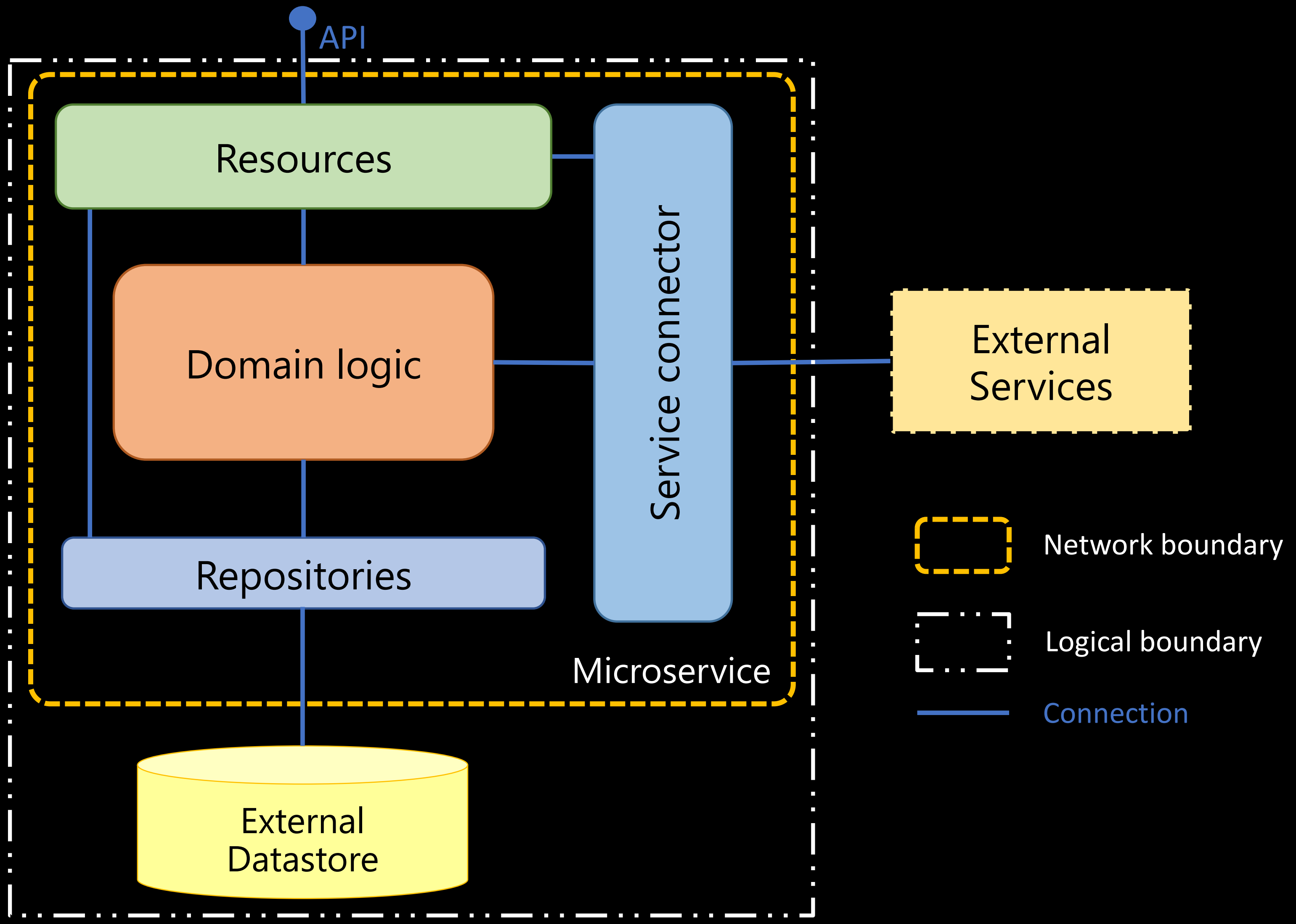
CouchDB

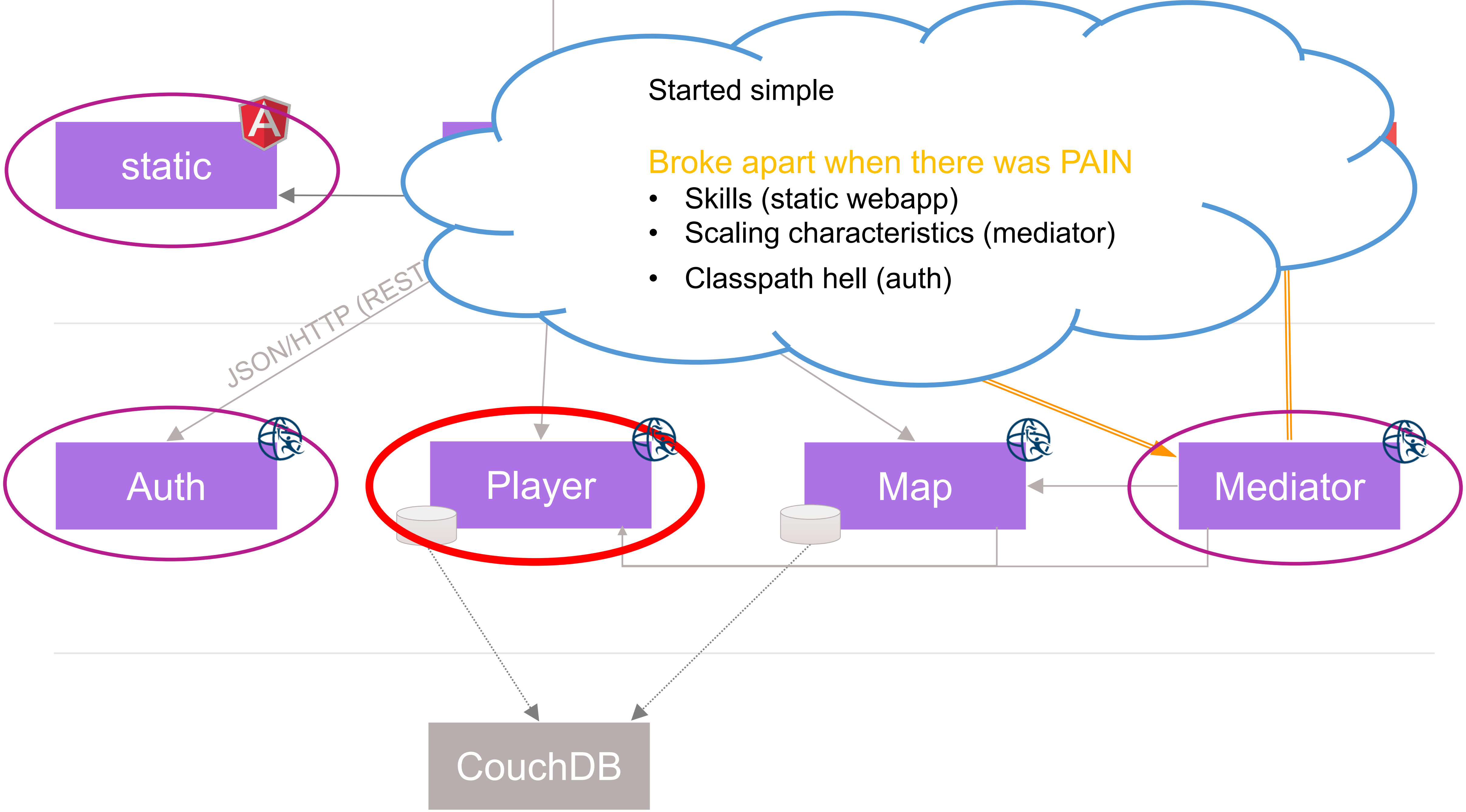
- CouchDB / Cloudant were better suited for Map
- For consistency/simplicity, we switched player over, too

The anatomy of a microservice

- Robustness principle
  - Be stingy in what you share
  - Be generous in what you accept

- Anti-corruption layers



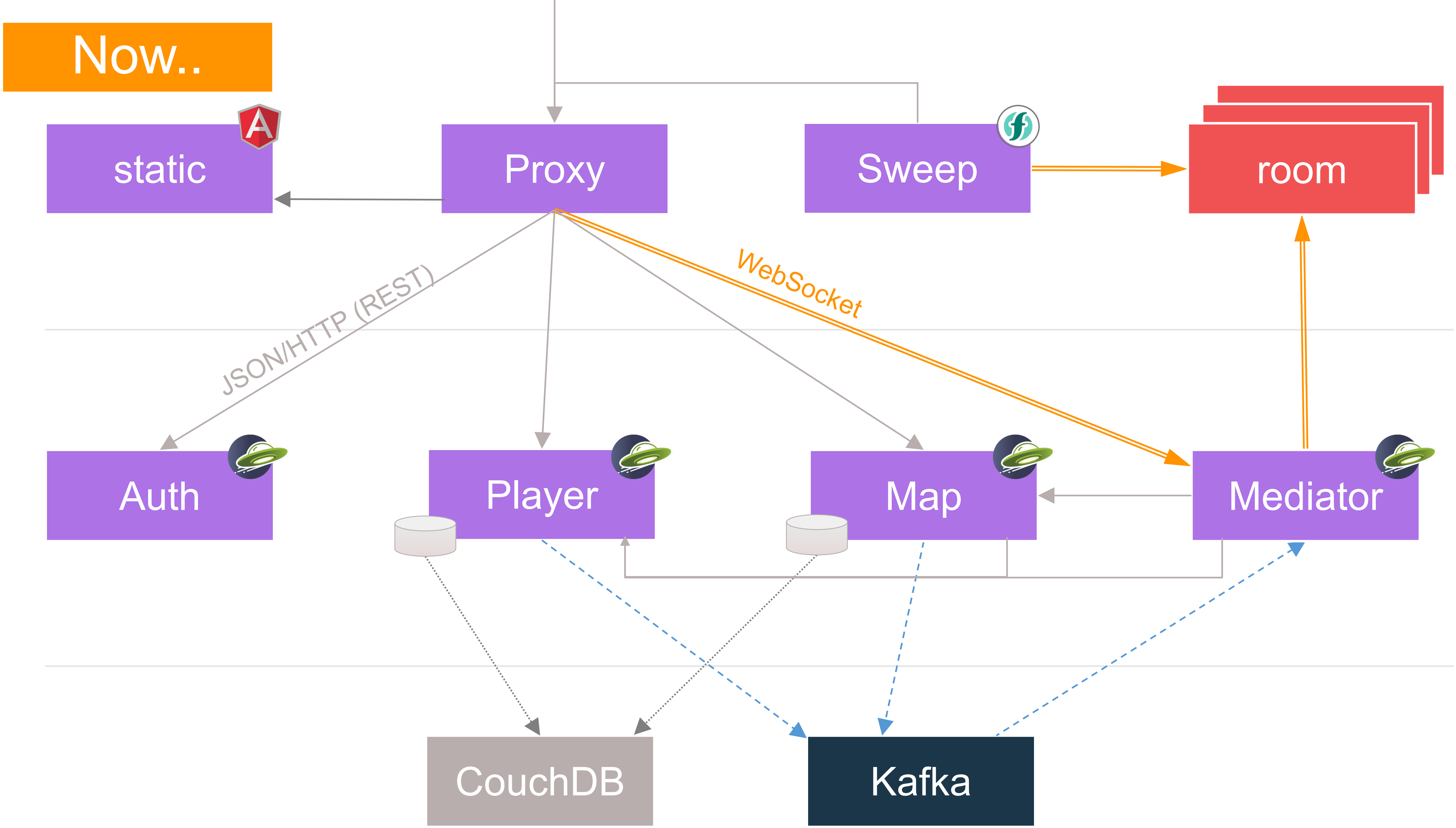


Started simple

Broke apart when there was PAIN

- Skills (static webapp)
- Scaling characteristics (mediator)
- Classpath hell (auth)

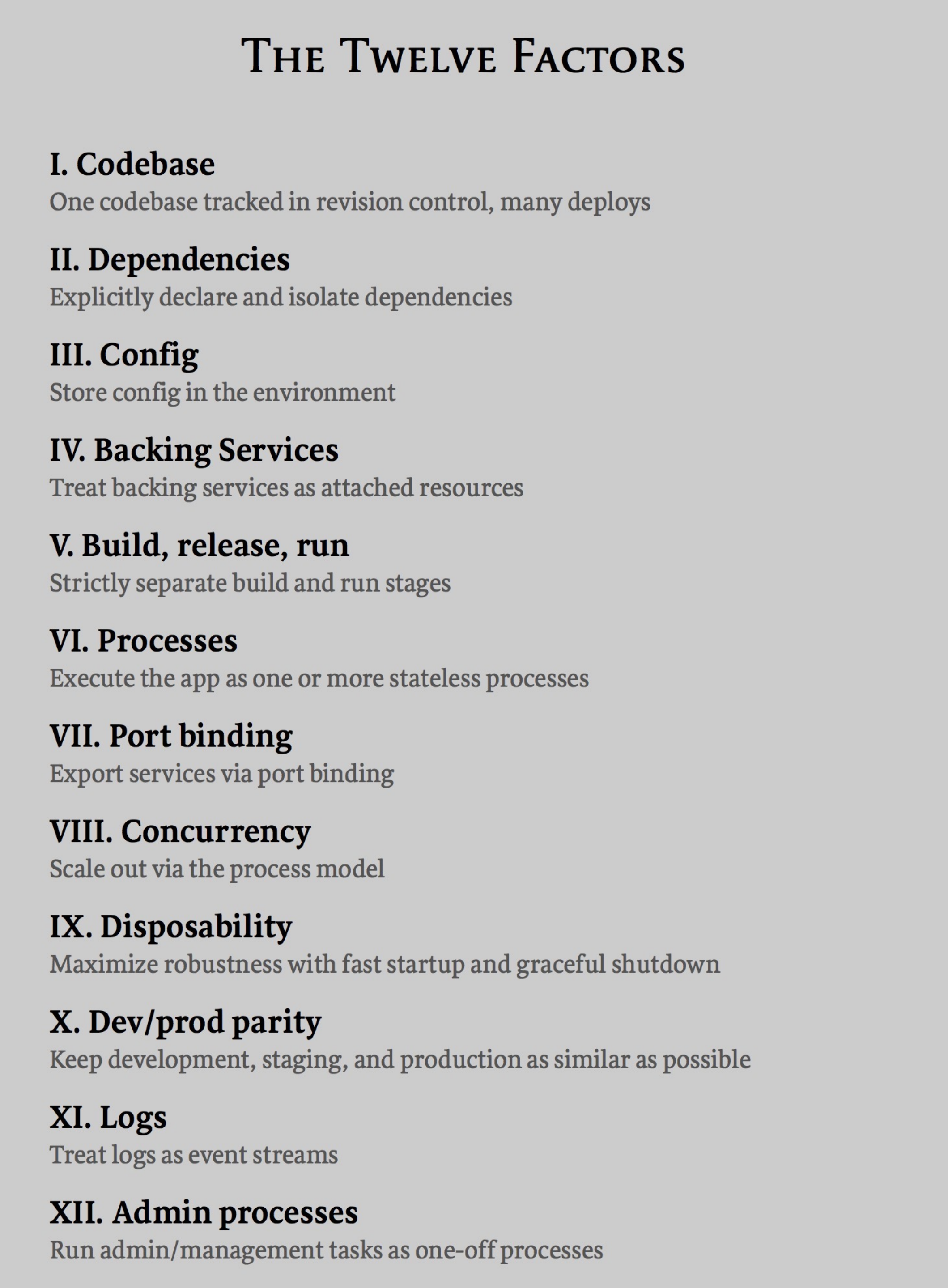
Now..


static
Proxy
Sweep
room
JSON/HTTP (REST)
WebSocket
Auth
Player
Map
Mediator
CouchDB
Kafka

Let’s make sure things are running..

```
$ go-run up       # if you haven’t already

$ go-run wait     # make sure things are good to go
  Game On! You're ready to play:  https://127.0.0.1


$ git clone https://github.com/stefanprodan/dockprom

$ cd dockprom

$ ADMIN_USER=admin ADMIN_PASSWORD=admin docker-compose up -d
```

Application vs. Environment



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

Local development

- Root /umbrella project for running core services locally
  - Scripts (go-admin.sh, go-run.sh)

- Local overrides for fast iteration
  - docker-compose.override.xml
  - Volume mounts

- Common base container for liberty image
  - Caching / Patching vulnerabilities

Let’s fix the web front end (1/2)

Click on the terms link: https://127.0.0.1

```
$ git submodule update --init webapp

$ cd webapp

$ ./build.sh
```

Using a container for build

- In gameon/webapp:
    - Dockerfile-node
    - docker/docker-build.sh (npm & gulp)
    - build.sh

- Key elements:
    - Specified user & group to avoid file permission issues
    - Specific volume for installed node modules
        - Avoid fighting over binaries between host and container

Let’s fix the web front end (2/2)

- In gameon/webapp directory:
  1. Open app/templates/default.html
  2. Around line 27, remove “templates/” before “terms.html”
- In gameon/docker directory:
  1. Copy docker-compose.override.yml.example → **docker-compose.override.yml**
  2. Open docker-compose.override.yml in editor of choice
  3. Uncomment webapp section

```
$ go-run rebuild webapp
```

→ Refresh browser!

Continuous Integration with Travis

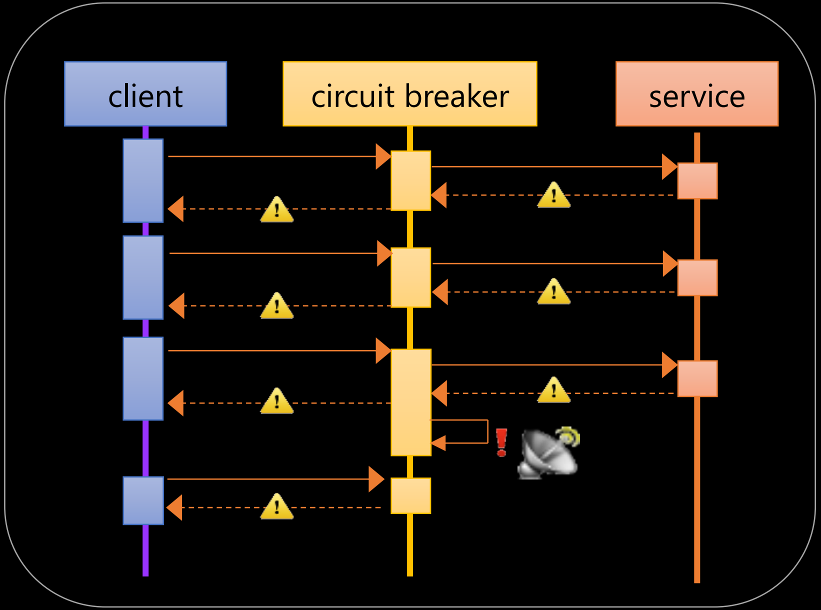
- Git submodules
  - Bridge between monolithic system and microservice
  - RULE: only CI/CD pipeline updates submodule versions

- Similar services use common build scripts
  - Common scripts stored in gameon/root repository
    - https://github.com/gameontext/gameon/tree/master/build
  - Simplifies maintenance
  - Git Ops – Everything for CI/CD is checked into and/or triggered by git

Fault Tolerance

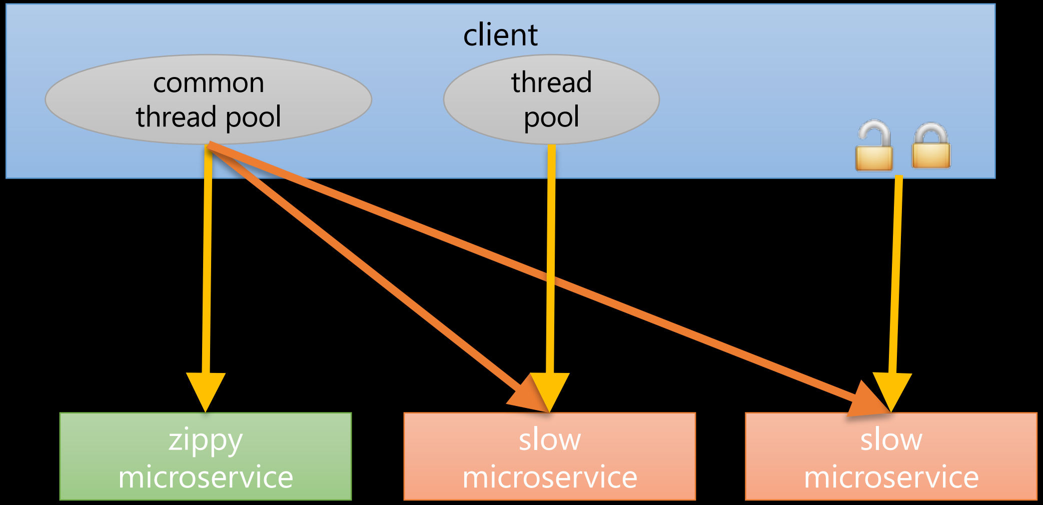
Circuit Breakers

- Wrap remote calls
- Monitor for failures
- Notify when circuit is tripped
- Retry or Fallback?
- When is circuit reset?



Bulkheads

- Ensure at most ‘n’ threads waiting for a slow resource
  - Thread isolation
    - With or without a queue
    - Timeout / fallback
  - Semaphore isolation
    - Request sent if lock obtained



Example of a fallback

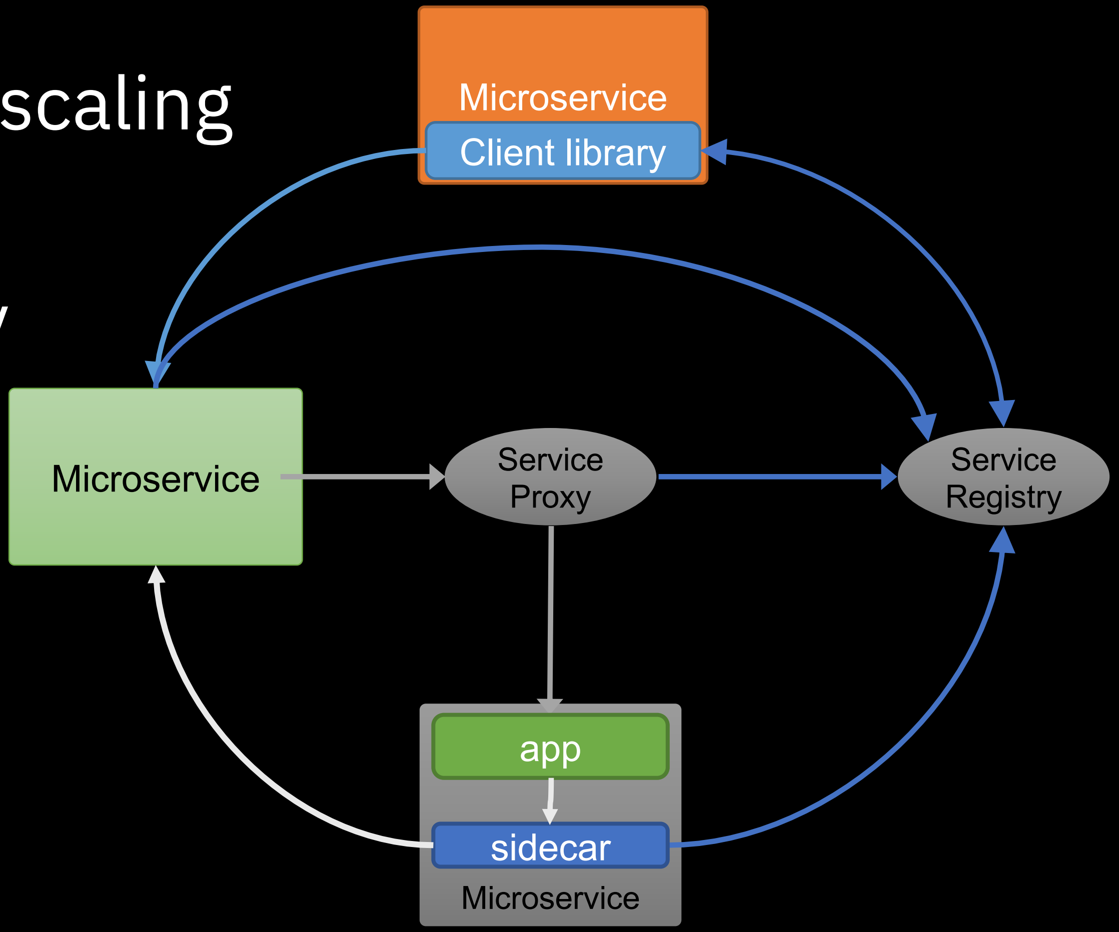
- /go <direction: E|W|N|S>
  - Repeat until you reach a “sick room”

- Examine things in the room:
  - What is the Mediator doing?
  - Would something in the infrastructure be able to do this?

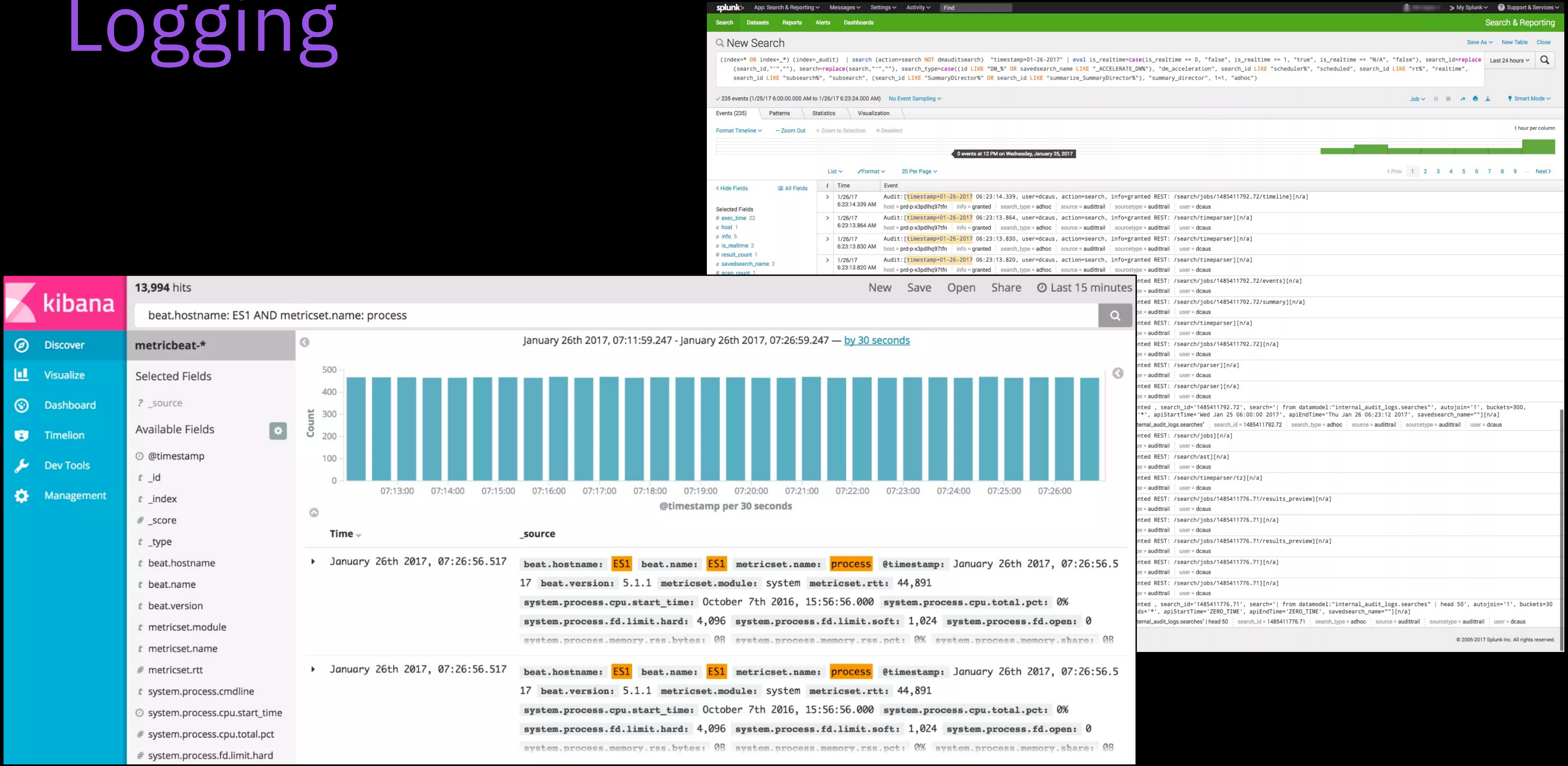
Service registration and discovery

- 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?



Logging



```
(index=* OR index=_*) (index=_audit)  | search (action=search NOT dmauditsearch)  "timestamp=01-26-2017" | eval is_realtime=case(is_realtime == 0, "false", is_realtime == 1, "true", is_realtime == "N/A", "false"), search_id=replace
    (search_id,"'",""), search=replace(search,"'",""), search_type=case((id LIKE "DM_%" OR savedsearch_name LIKE "_ACCELERATE_DM%"), "dm_acceleration", search_id LIKE "scheduler%", "scheduled", search_id LIKE "rt%", "realtime",
    search_id LIKE "subsearch%", "subsearch", (search_id LIKE "SummaryDirector%" OR search_id LIKE "summarize_SummaryDirector%"), "summary_director", 1=1, "adhoc")
```

✓ 235 events (1/25/17 6:00:00.000 AM to 1/26/17 6:23:24.000 AM)

0 events at 12 PM on Wednesday, January 25, 2017

```
i  Time                    Event
>  1/26/17                 Audit:[timestamp=01-26-2017 06:23:14.339, user=dcaus, action=search, info=granted REST: /search/jobs/1485411792.72/timeline][n/a]
   6:23:14.339 AM          host = prd-p-x3pdlhq97tfn   info = granted   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
>  1/26/17                 Audit:[timestamp=01-26-2017 06:23:13.864, user=dcaus, action=search, info=granted REST: /search/timeparser][n/a]
   6:23:13.864 AM          host = prd-p-x3pdlhq97tfn   info = granted   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
>  1/26/17                 Audit:[timestamp=01-26-2017 06:23:13.830, user=dcaus, action=search, info=granted REST: /search/timeparser][n/a]
   6:23:13.830 AM          host = prd-p-x3pdlhq97tfn   info = granted   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
>  1/26/17                 Audit:[timestamp=01-26-2017 06:23:13.820, user=dcaus, action=search, info=granted REST: /search/timeparser][n/a]
   6:23:13.820 AM          host = prd-p-x3pdlhq97tfn   info = granted   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
nted REST: /search/jobs/1485411792.72/events][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411792.72/summary][n/a]
pe = audittrail   user = dcaus
nted REST: /search/timeparser][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411792.72][n/a]
pe = audittrail   user = dcaus
nted REST: /search/parser][n/a]
pe = audittrail   user = dcaus
nted REST: /search/parser][n/a]
pe = audittrail   user = dcaus
nted , search_id='1485411792.72', search='| from datamodel:"internal_audit_logs.searches"', autojoin='1', buckets=300,
'*', apiStartTime='Wed Jan 25 06:00:00 2017', apiEndTime='Thu Jan 26 06:23:12 2017', savedsearch_name=""][n/a]
ternal_audit_logs.searches"   search_id = 1485411792.72   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
nted REST: /search/jobs][n/a]
pe = audittrail   user = dcaus
nted REST: /search/ast][n/a]
pe = audittrail   user = dcaus
nted REST: /search/timeparser/tz][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411776.71/results_preview][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411776.71][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411776.71/results_preview][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411776.71][n/a]
pe = audittrail   user = dcaus
nted REST: /search/jobs/1485411776.71][n/a]
pe = audittrail   user = dcaus
nted , search_id='1485411776.71', search='| from datamodel:"internal_audit_logs.searches" | head 50', autojoin='1', buckets=30
ds='*', apiStartTime='ZERO_TIME', apiEndTime='ZERO_TIME', savedsearch_name=""][n/a]
ternal_audit_logs.searches" | head 50   search_id = 1485411776.71   search_type = adhoc   source = audittrail   sourcetype = audittrail   user = dcaus
```

© 2005-2017 Splunk Inc. All rights reserved.

13,994 hits

beat.hostname: ES1 AND metricset.name: process

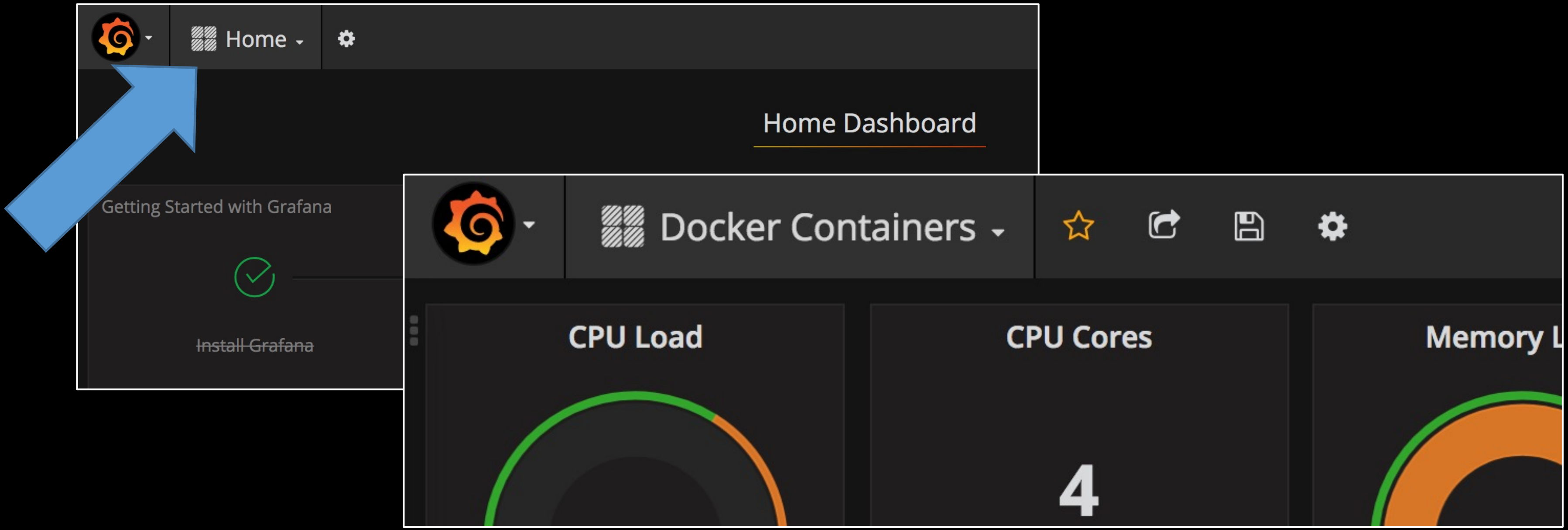
January 26th 2017, 07:11:59.247 - January 26th 2017, 07:26:59.247 — by 30 seconds

```
  Time ▾                             _source
▸ January 26th 2017, 07:26:56.517   beat.hostname: ES1 beat.name: ES1 metricset.name: process @timestamp: January 26th 2017, 07:26:56.5
                                    17 beat.version: 5.1.1 metricset.module: system metricset.rtt: 44,891
                                    system.process.cpu.start_time: October 7th 2016, 15:56:56.000 system.process.cpu.total.pct: 0%
                                    system.process.fd.limit.hard: 4,096 system.process.fd.limit.soft: 1,024 system.process.fd.open: 0
                                    system.process.memory.rss.bytes: 0B system.process.memory.rss.pct: 0% system.process.memory.share: 0B
▸ January 26th 2017, 07:26:56.517   beat.hostname: ES1 beat.name: ES1 metricset.name: process @timestamp: January 26th 2017, 07:26:56.5
                                    17 beat.version: 5.1.1 metricset.module: system metricset.rtt: 44,891
                                    system.process.cpu.start_time: October 7th 2016, 15:56:56.000 system.process.cpu.total.pct: 0%
                                    system.process.fd.limit.hard: 4,096 system.process.fd.limit.soft: 1,024 system.process.fd.open: 0
                                    system.process.memory.rss.bytes: 0B system.process.memory.rss.pct: 0% system.process.memory.share: 0B
```

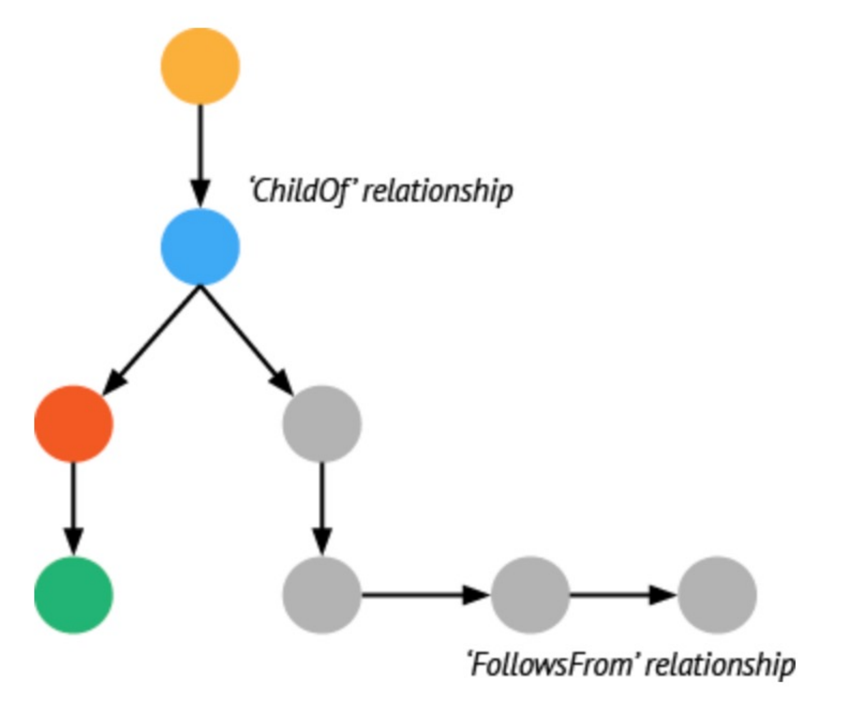
Metrics

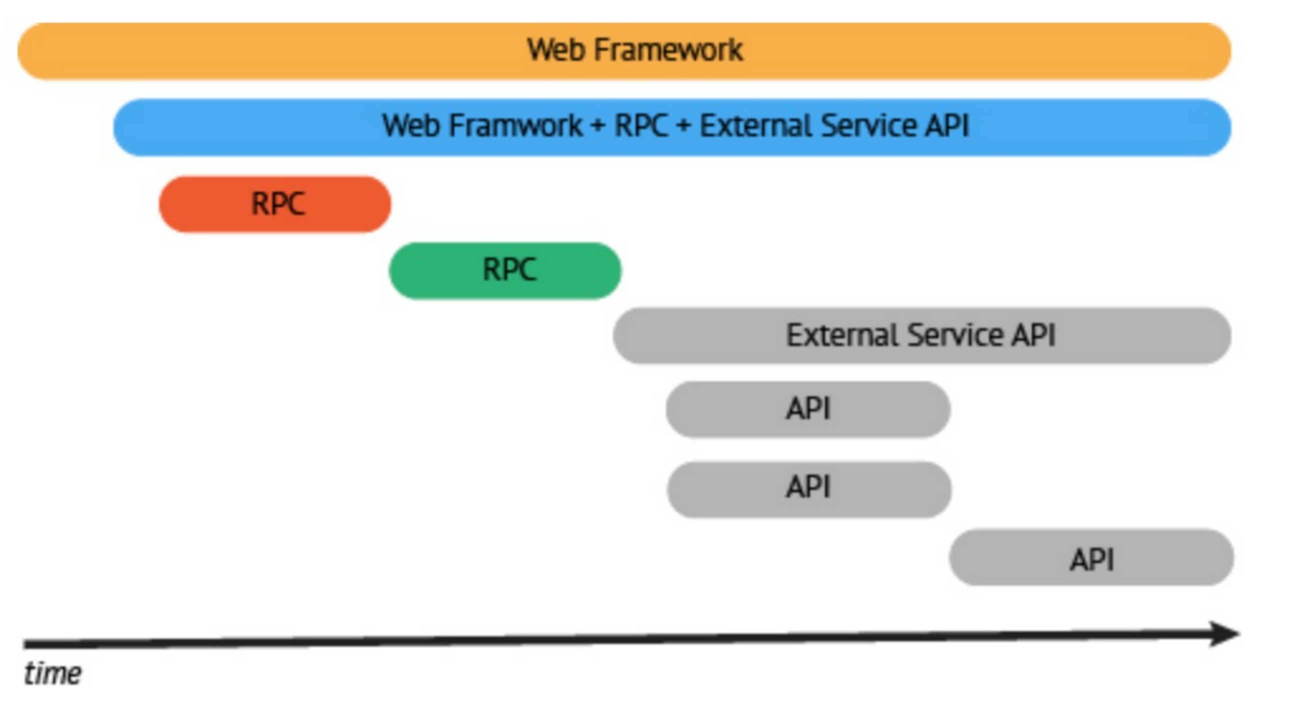
Go to: http://127.0.0.1:3000

Log in with admin / admin



Open Tracing





References

- Game On!
  - https://gameontext.org/#/
  - Evolution of the Architecture: https://book.gameontext.org/chronicles/
- IBM Redbooks: Microservices Best Practices for Java http://www.redbooks.ibm.com/abstracts/sg248357.html?Open
- Toby Clemson: Testing Strategies in a Microservice Architecture https://martinfowler.com/articles/microservice-testing/#anatomy-modules
- Istio: https://istio.io/
  - Istio Distributed Tracing: https://istio.io/docs/tasks/telemetry/distributed-tracing.html
- Kubernetes: https://kubernetes.io/
  - https://brancz.com/2018/01/05/prometheus-vs-heapster-vs-kubernetes-metrics-apis/

Notices and disclaimers

-
- **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 per 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 follows 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 about 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 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 and [names of other referenced IBM products and services used in the presentation] 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.