

Exploring Terracotta

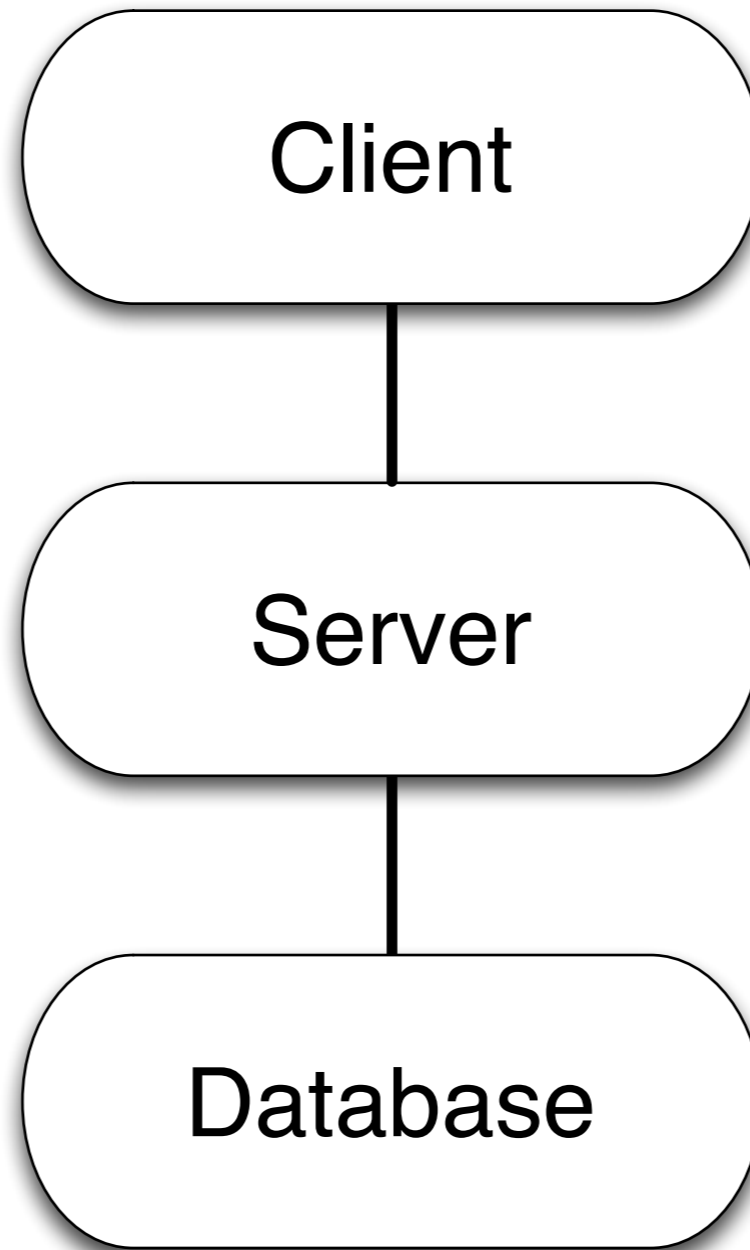
Alex Miller (@puredanger)



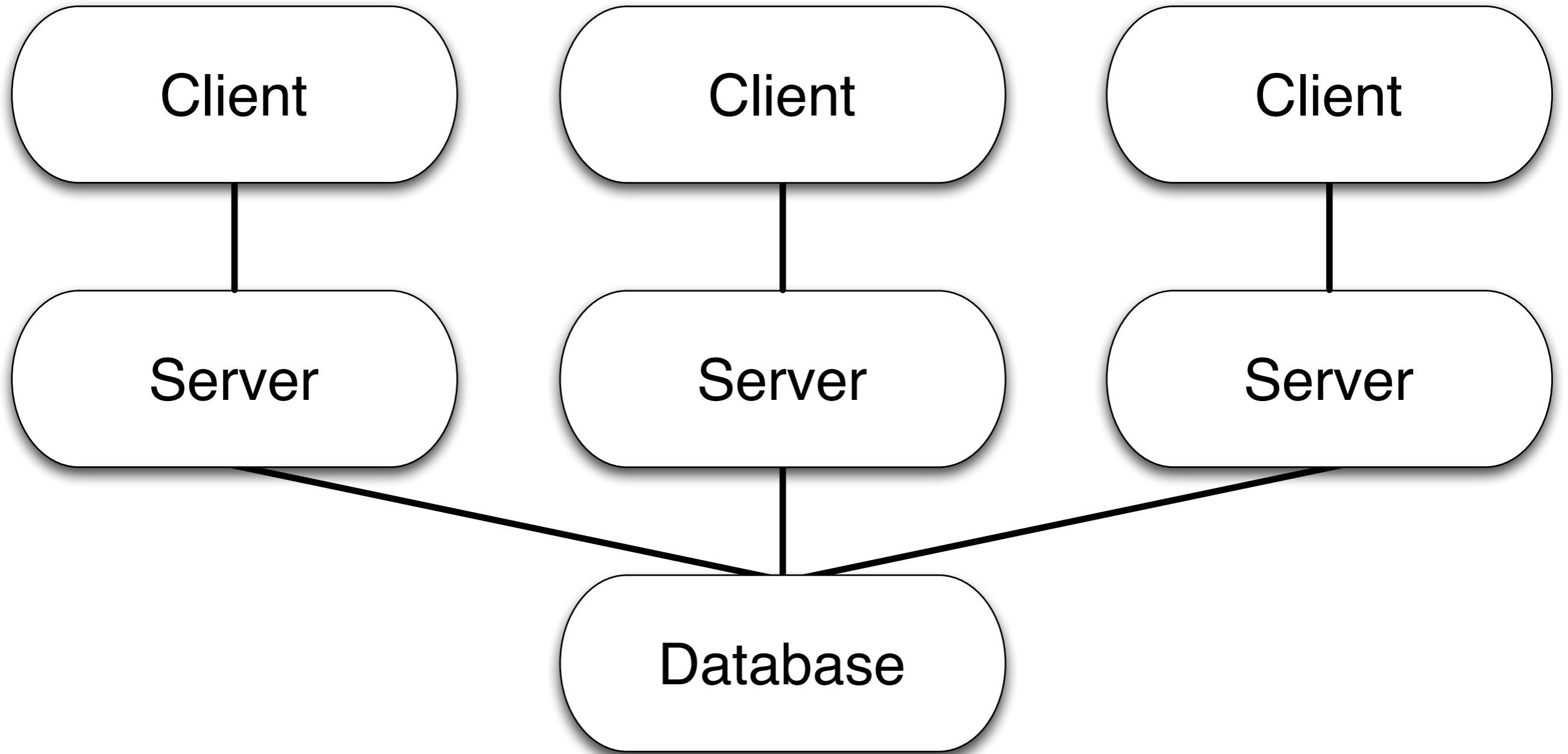
Agenda

- **Architecture and scale**
- Examiner reference application
- What do people use it for?
- How does it work?
- Terracotta 3.0

Applications in Three Tiers

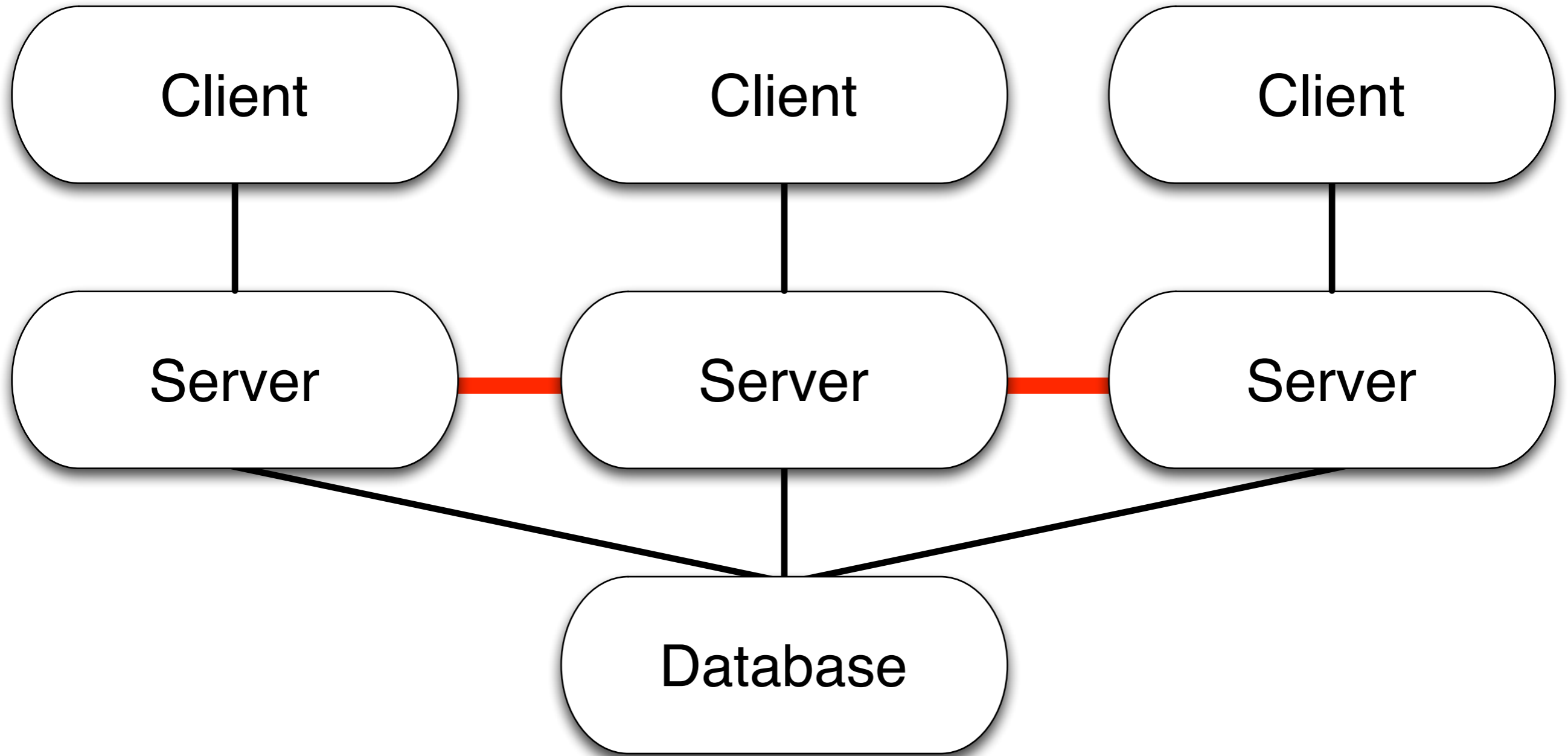


How do we scale?



But what about failover?

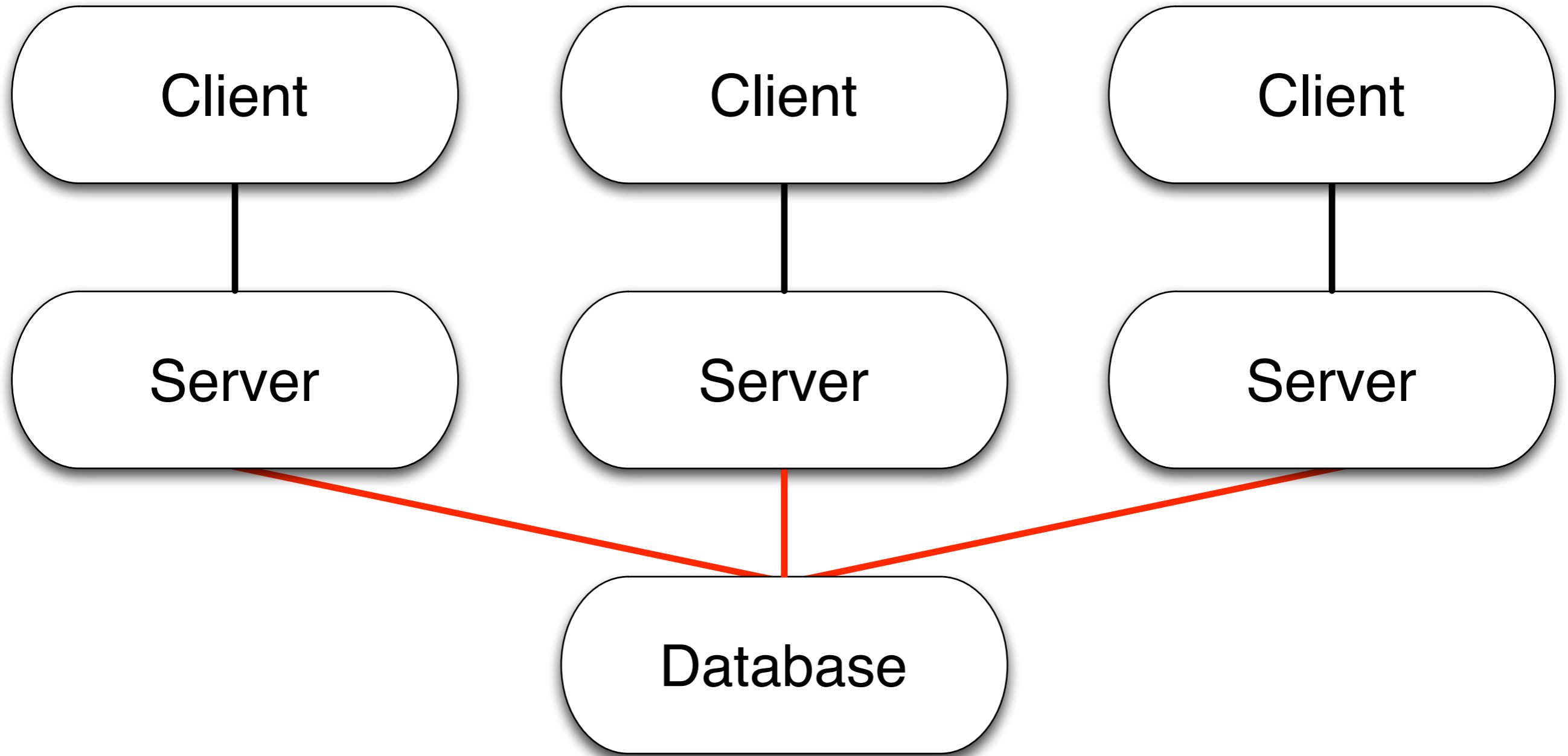
Option 1: Servers share state



How?

- RMI
- JMS
- Custom (JGroups, etc)

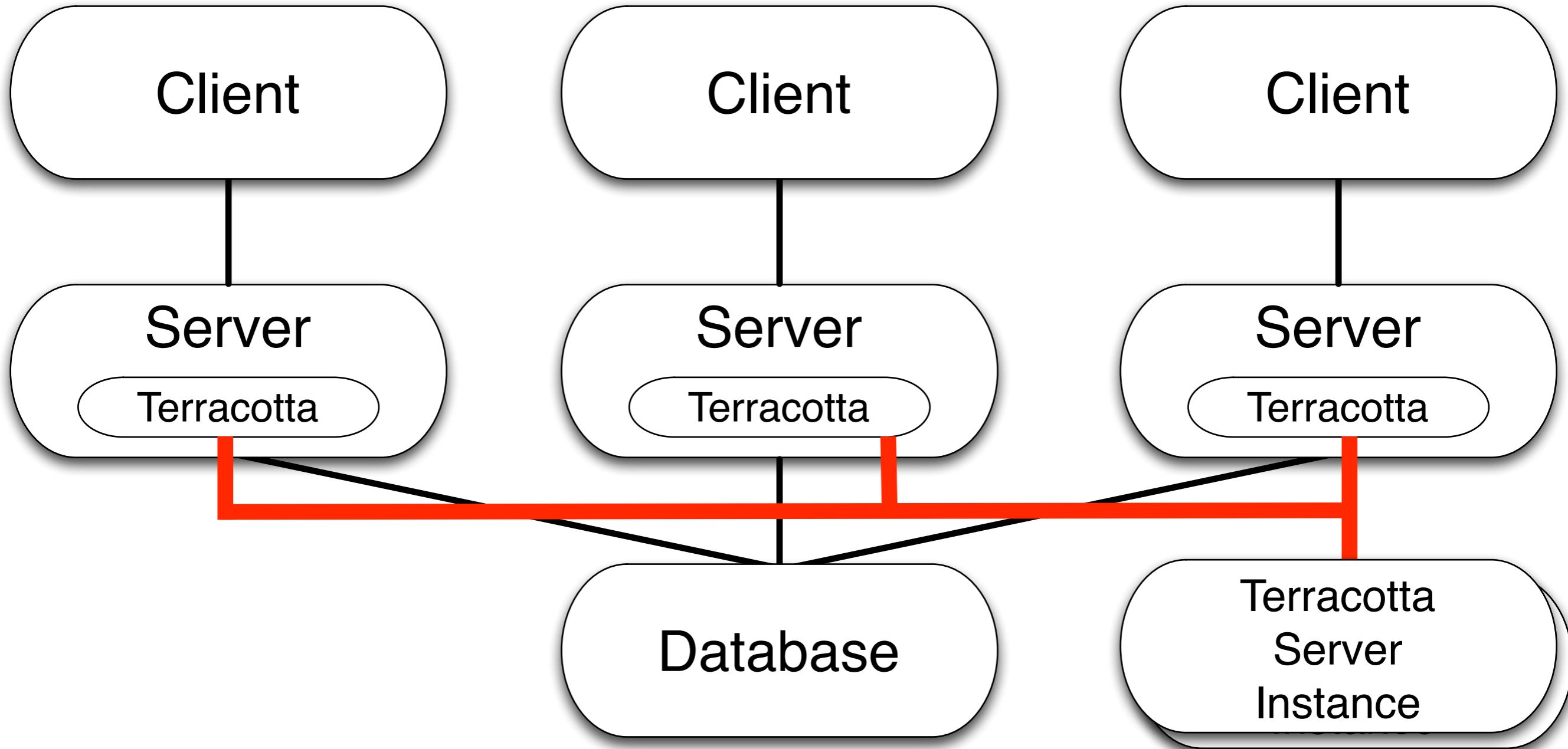
Option 2: Database (“stateless”)



Stateless == “State in the database”

- Database load
- Serialization to and from the database
- Network bandwidth
- Object / relational mapping due to impedance mismatch

Option 3: Terracotta!



Why is this better?

- Hub and spoke vs peer to peer
- Avoid object / relational translation
- Avoid serialization
- Reduce database overload
- Programming model you already know
- Focus on your app, not scalability and high availability

Agenda

- Architecture and scale
- **Examinator reference application**
- What do people use it for?
- How does it work?
- Terracotta 3.0

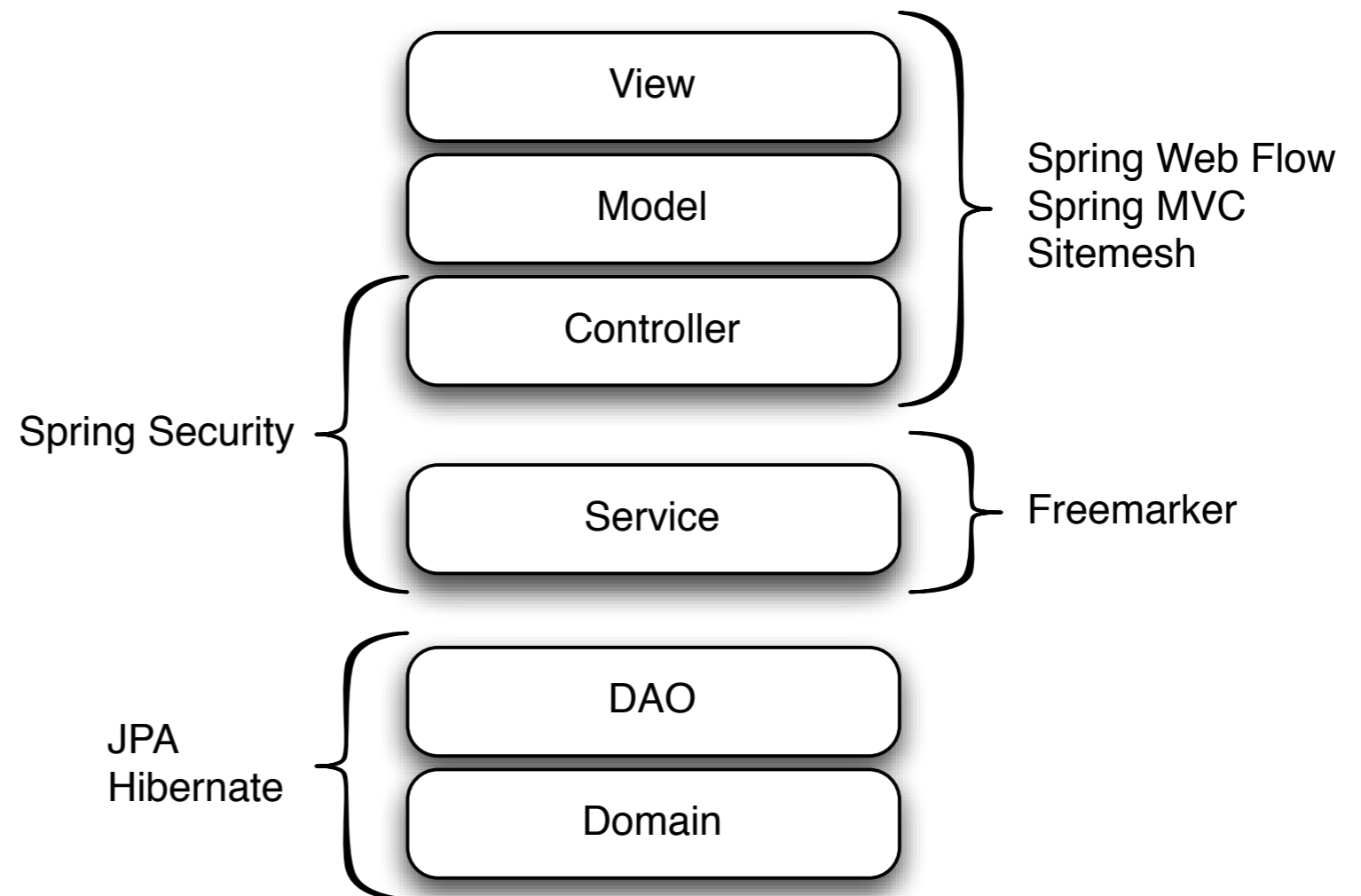
Examinator

- Terracotta reference web application
 - Online exams
 - “Session” use case
-
- <http://reference.terracotta.org>

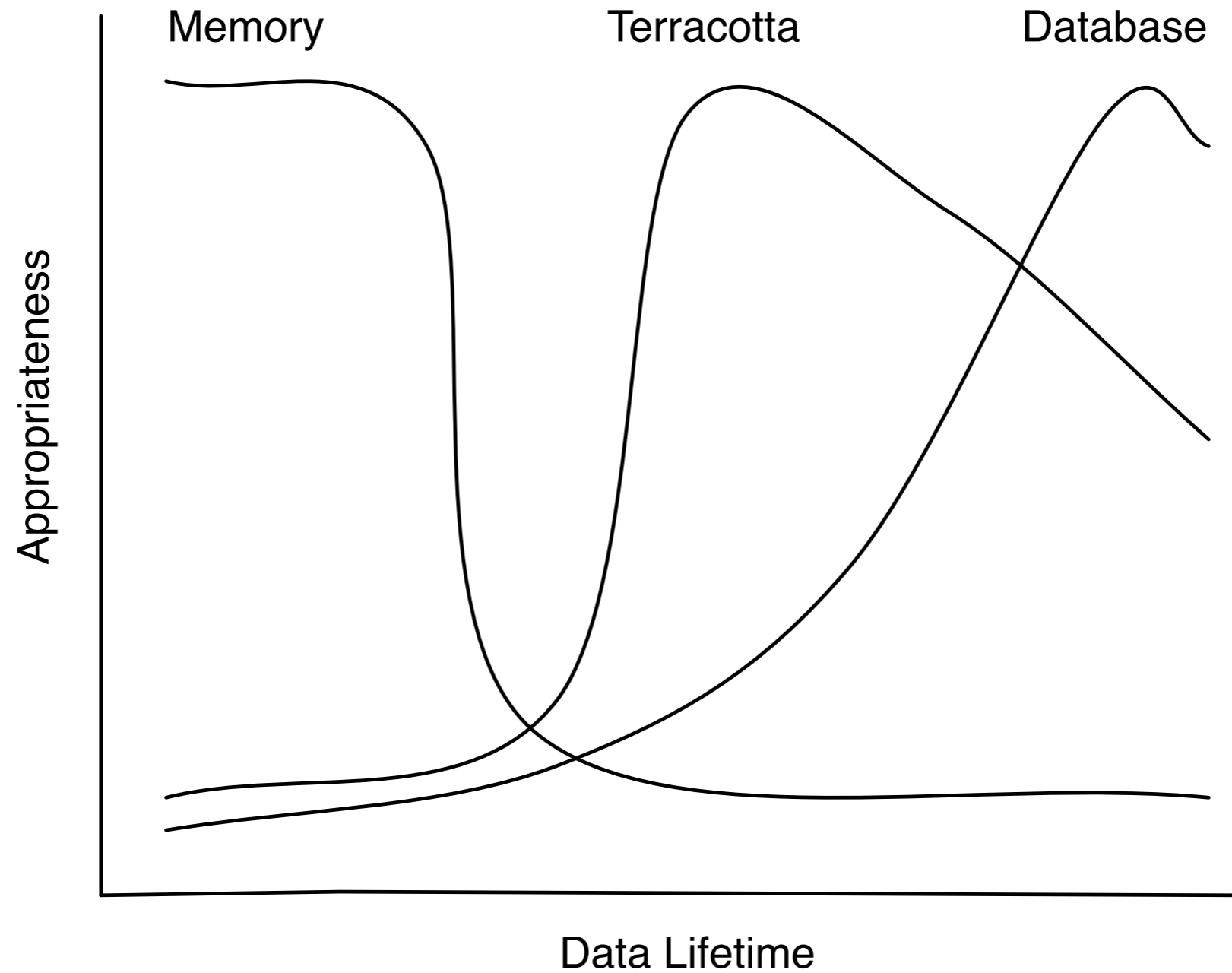


Examinator stack

- Spring-based stack
 - MVC
 - Web Flow
 - Security
 - Transactions
- Open source
 - Tomcat / Jetty
 - Hibernate / JPA
 - Site Mesh
 - Freemarker



What state should go in Terracotta?



Examinator Terracotta Usage

- User registration codes
- Password reset codes
- Exam cache
- Session clustering
- Spring security (via session)
- Spring web flow (via session)
- **Exam in progress session** - answers, choice ordering, marked to review

Demo

Agenda

- Architecture and scale
- Examiner reference application
- **What do people use it for?**
- How does it work?
- Terracotta 3.0

Business Use Cases

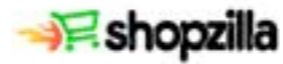
- Distributed cache
- Clustered HTTP sessions
- Batch processing
- Grid
- Messaging and events

Terracotta Elements

- Replicated sessions
- Async processor
- Cache evictor
- ConcurrentStringMap
- Queues
- Master-worker
- TC Cache

Who Uses It?

- e-Commerce
- Online gaming
- Financial services
- Travel & leisure
- Social networking



Agenda

- Architecture and scale
- Examiner reference application
- What do people use it for?
- **How does it work?**
- Terracotta 3.0

Core Concepts

- Roots
- Instrumented Classes
- Locks
- Integration Modules

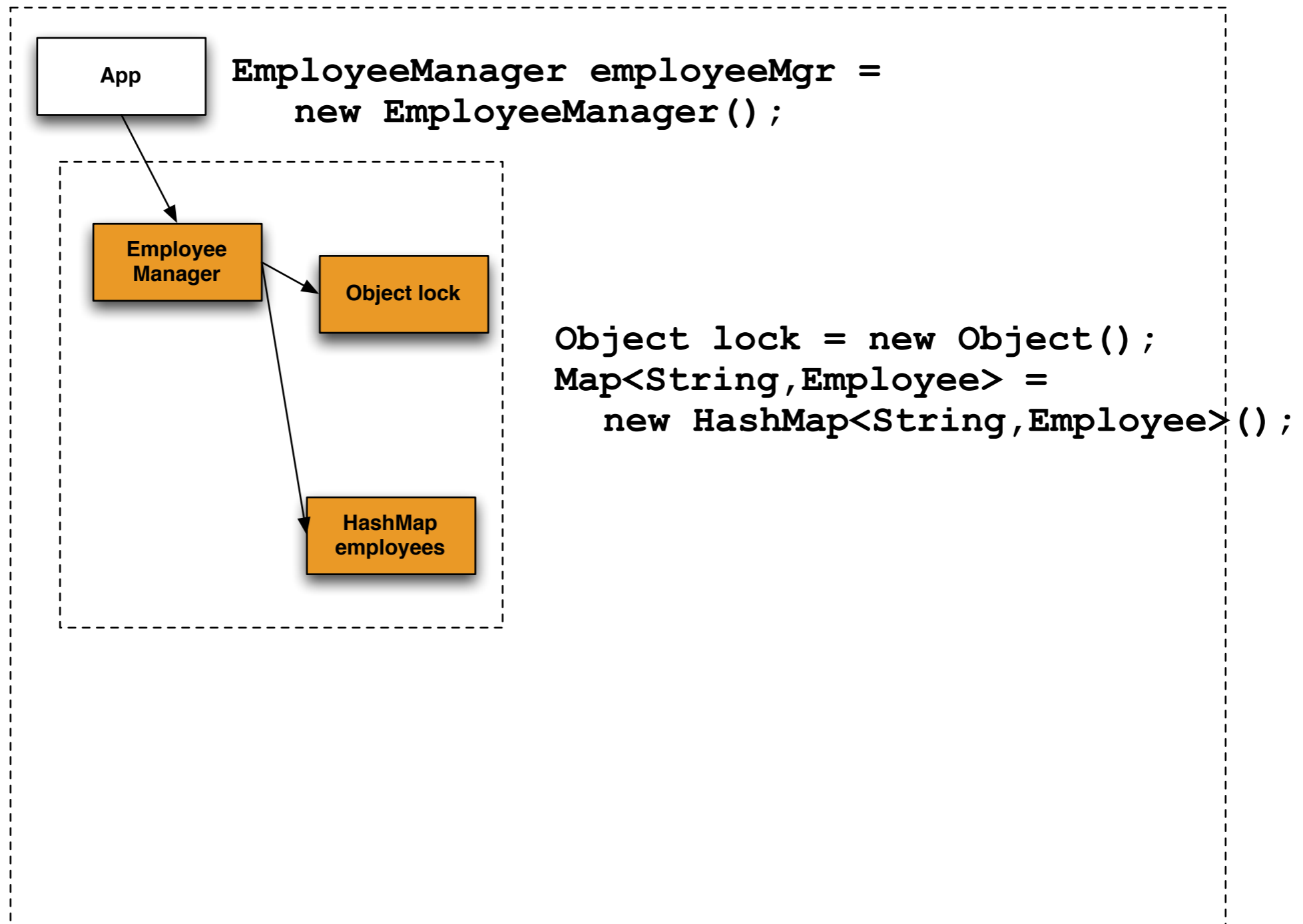
Clustered Heap



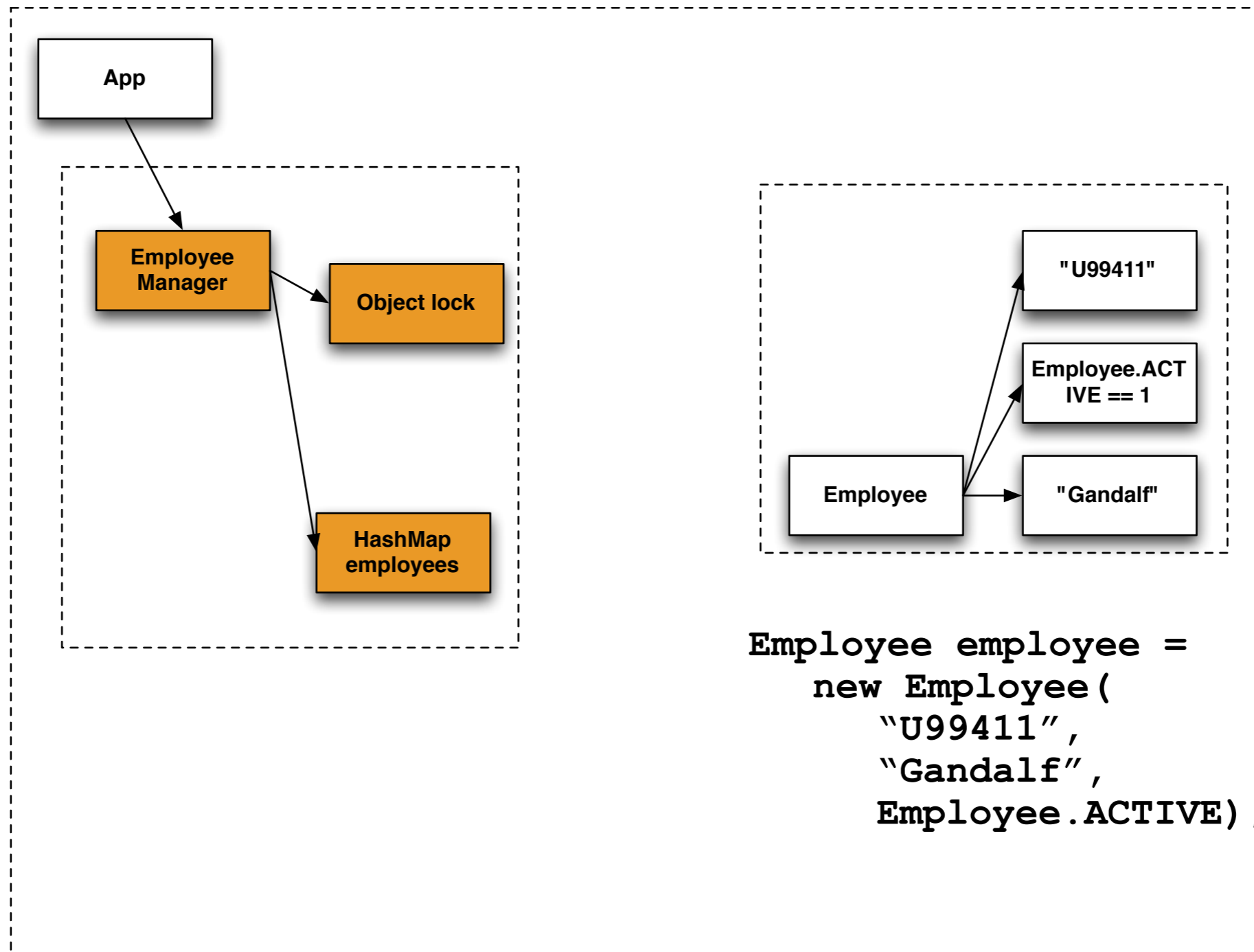
App

```
App app = new App ();  
app.run ();
```

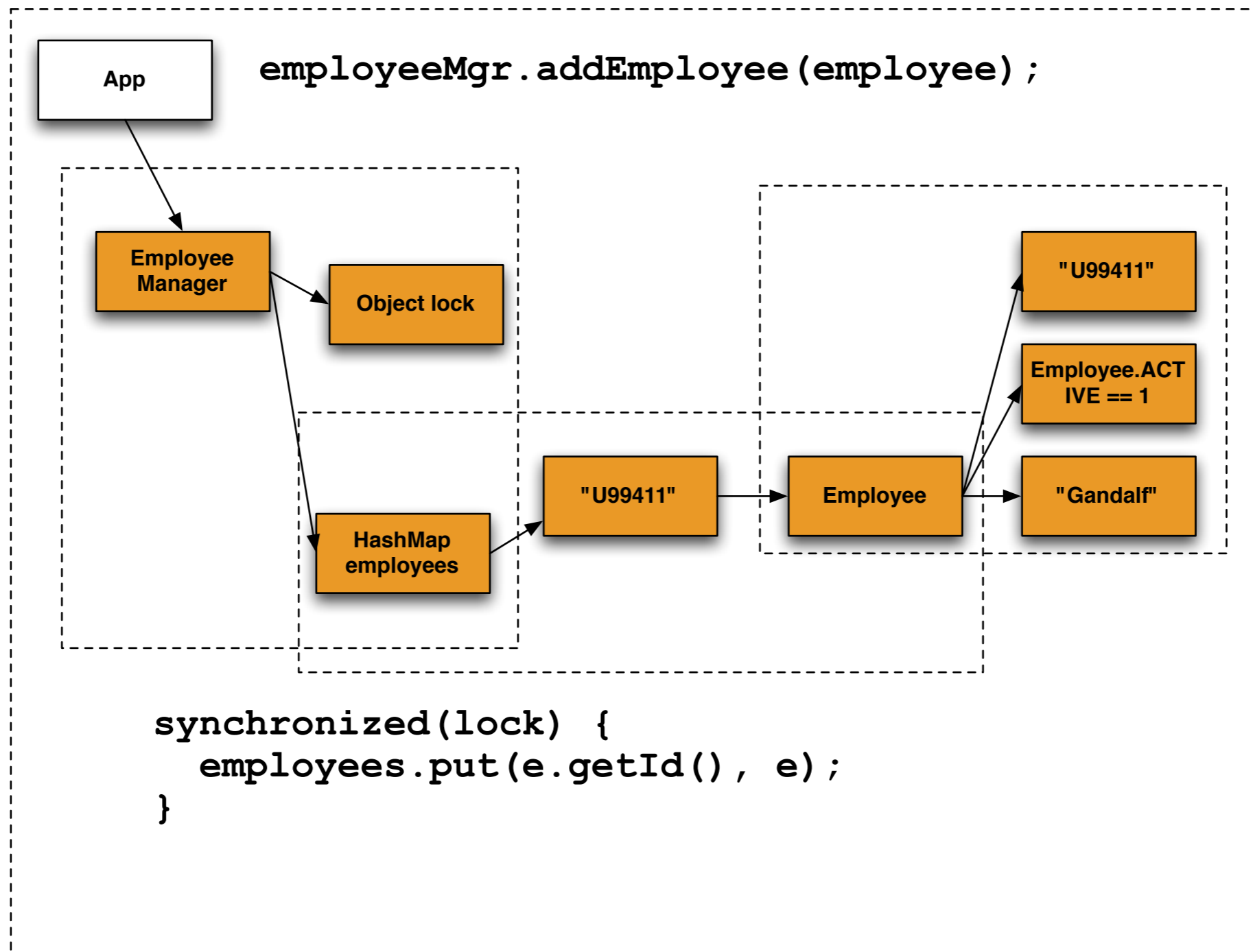
Clustered Heap



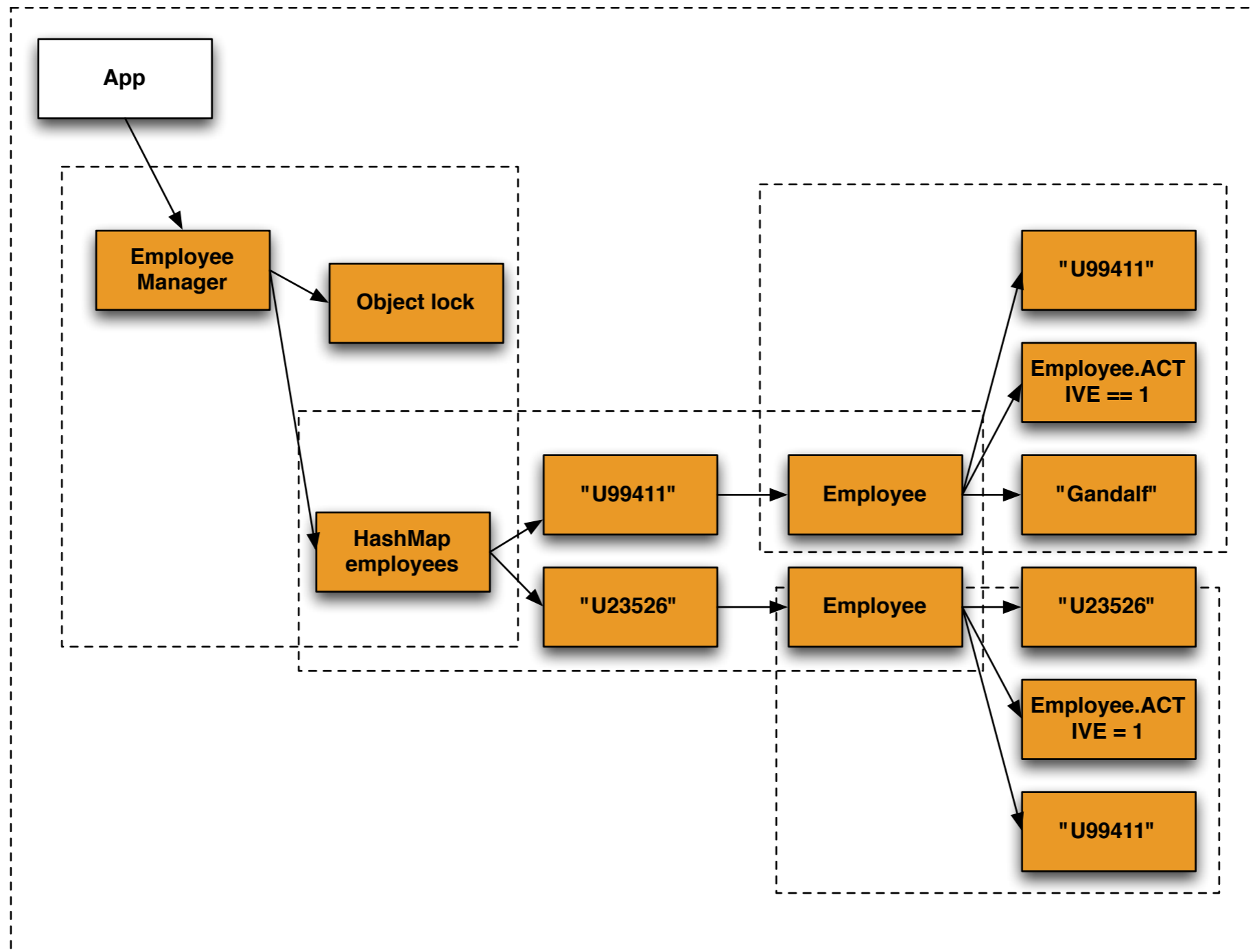
Clustered Heap



Clustered Heap



Clustered Heap



Let's look at how to do that...

Agenda

- Architecture and scale
- Examiner reference application
- What do people use it for?
- How does it work?
- **Terracotta 3.0**

Terracotta 3.0

- Server Array Striping - server scalability
- Operations Center / Developer Console - new dashboard, DGC stats
- Cluster events API - topology, data locality
- Application groups - sharing data across heterogeneous applications
- Platform support - Oracle WebLogic 10.2, 10.3 and RHEL 5.0

Thanks!

- Terracotta Open Source JVM clustering:
 - <http://www.terracotta.org>
- Apress: “The Definitive Guide to Terracotta”
 - by Ari Zilka, Alex Miller, Geert Bevin, Jonas Boner, Orion Letizi, Taylor Gautier
- Alex Miller
 - @puredanger
 - <http://tech.puredanger.com>

