



**Mathieu ANCELIN
Chris WOODROW
Alexandre DELEGUE**





Mathieu ANCELIN

- Développeur @SERLI
- Scala, Java, web & OSS
 - ReactiveCouchbase, Weld-OSGi, Weld, etc ...
 - Poitou-Charentes JUG
- Membre de l'expert group CDI 1.1 (JSR-346)
- Membre de l'expert group OSGi Enterprise
- @TrevorReznik





Chris Woodrow

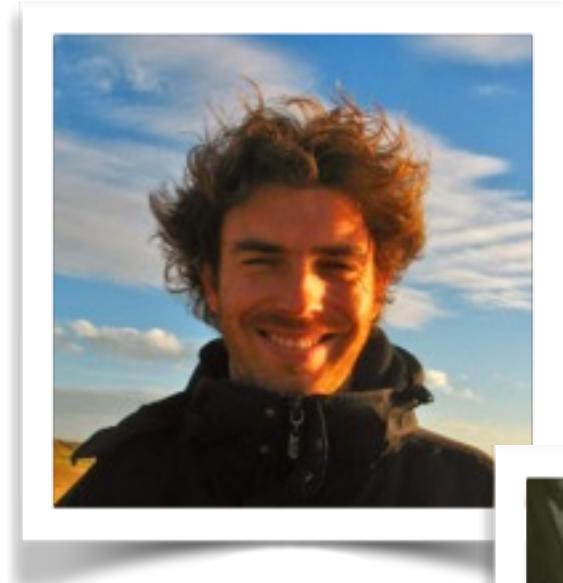
- Développeur @Serli
 - Java
 - Cloud
 - Cassandra
 - ElasticSearch
 - Haute Disponibilité
- InfoQ.FR





Alexandre DELEGUE

- Développeur @ SERLI
 - Java
 - Scala
 - Web
 - spring, play, ...
- @chanksleroux



- Société de conseil et d'ingénierie du SI
- 75 personnes
- 80% de business Java
- Contribution à des projets OSS
- 10% de la force de travail sur l'OSS
- Membre de l'EG JSR-346
- Membre de l'OSGi Alliance
- www.serli.com @SerliFr



SERLI

- Société de conseil et d'ingénierie du SI
 - 75 personnes
 - 80% de business Java
 - Contribution à des projets OSS
 - 10% de la force de travail sur l'OSS
 - Membre de l'EG JSR-346
 - Membre de l'OSGi Alliance
 - www.serli.com @SerliFr



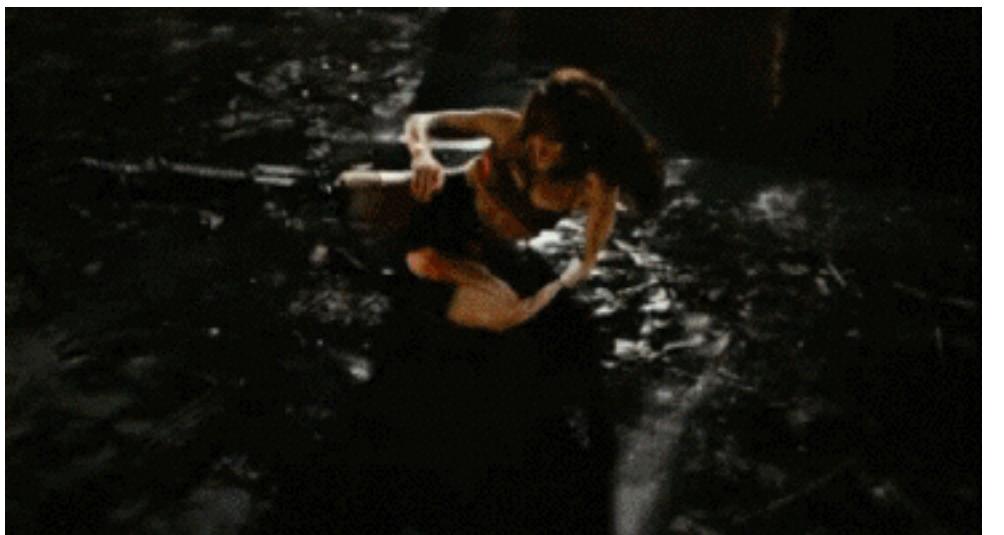
SERLi

Les technologies présentées sont inspirées de technologies réelles

Les applications qui en découlent sont fictives
Toute ressemblance avec des applications existantes n'est que fortuite



SERLi



SERLi

www.amazing.com



Amazing Store

Hello test

0 items

Logout



Search a product

Search



Mitraillette années 30 canon court

Mitraillette années 30. S ...

3599.0 €

Add to cart

Mitraillette années 30

Mitraillette années 30. P ...

3599.0 €

Add to cart

Arbalète et accessoires

Magnifique arbalète pouli ...

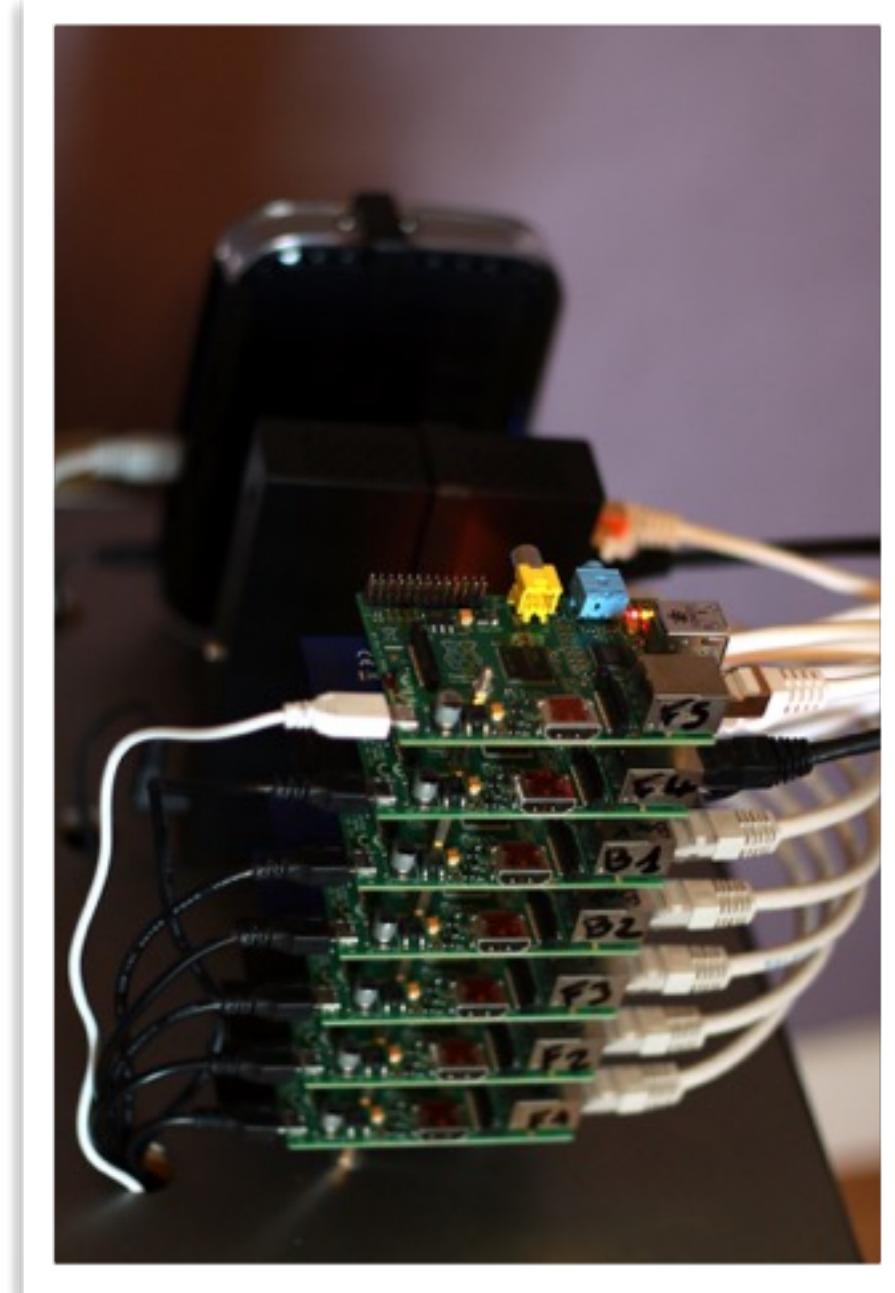
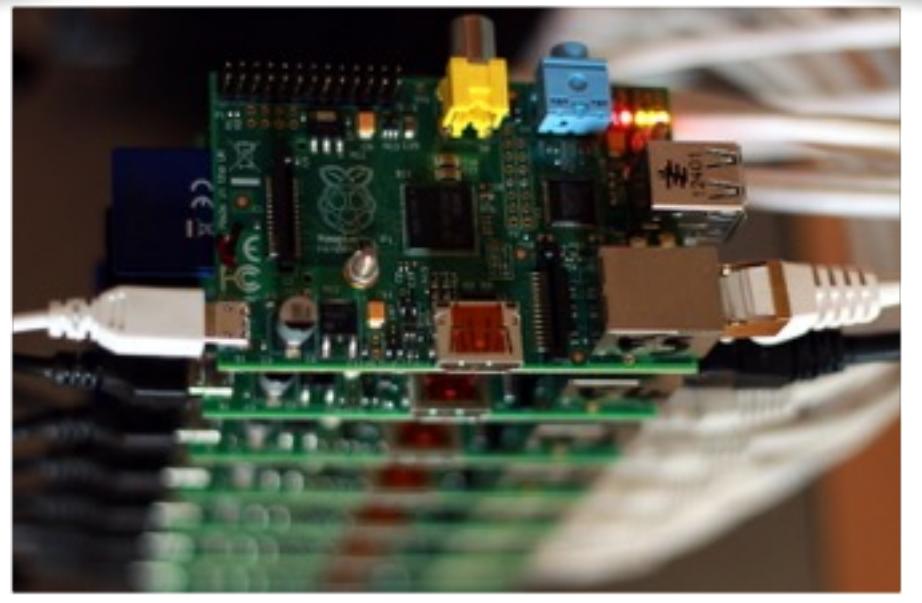
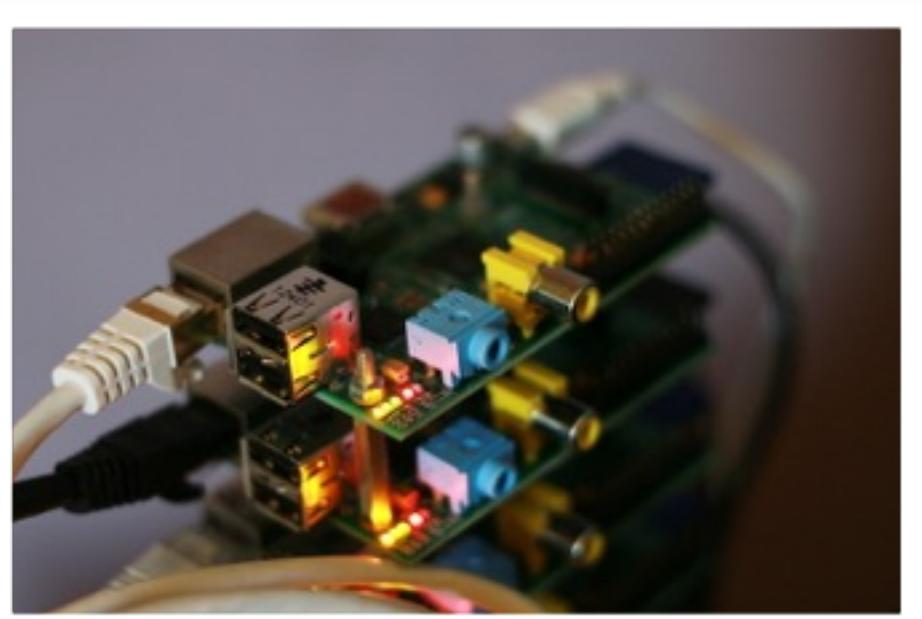
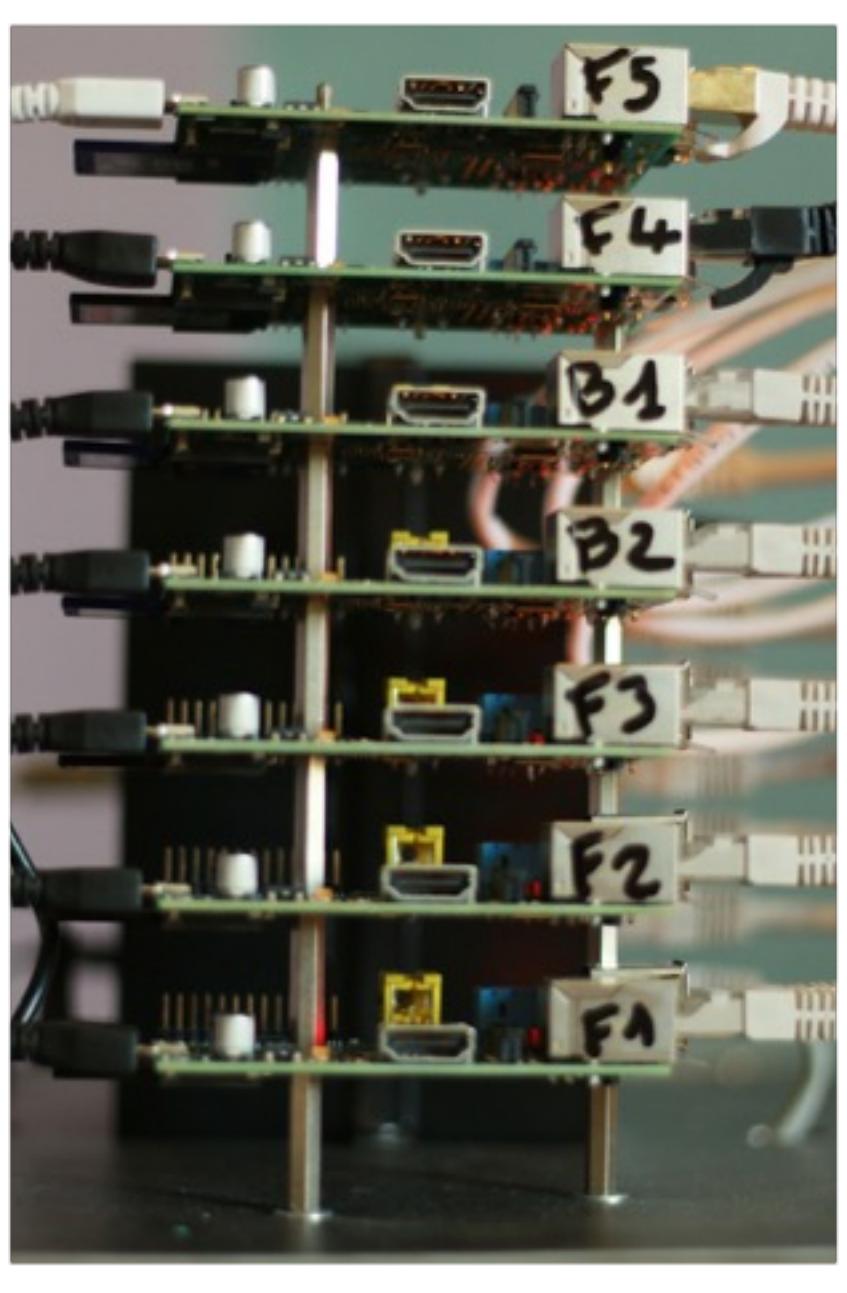
200.99 €

Add to cart

SERLi



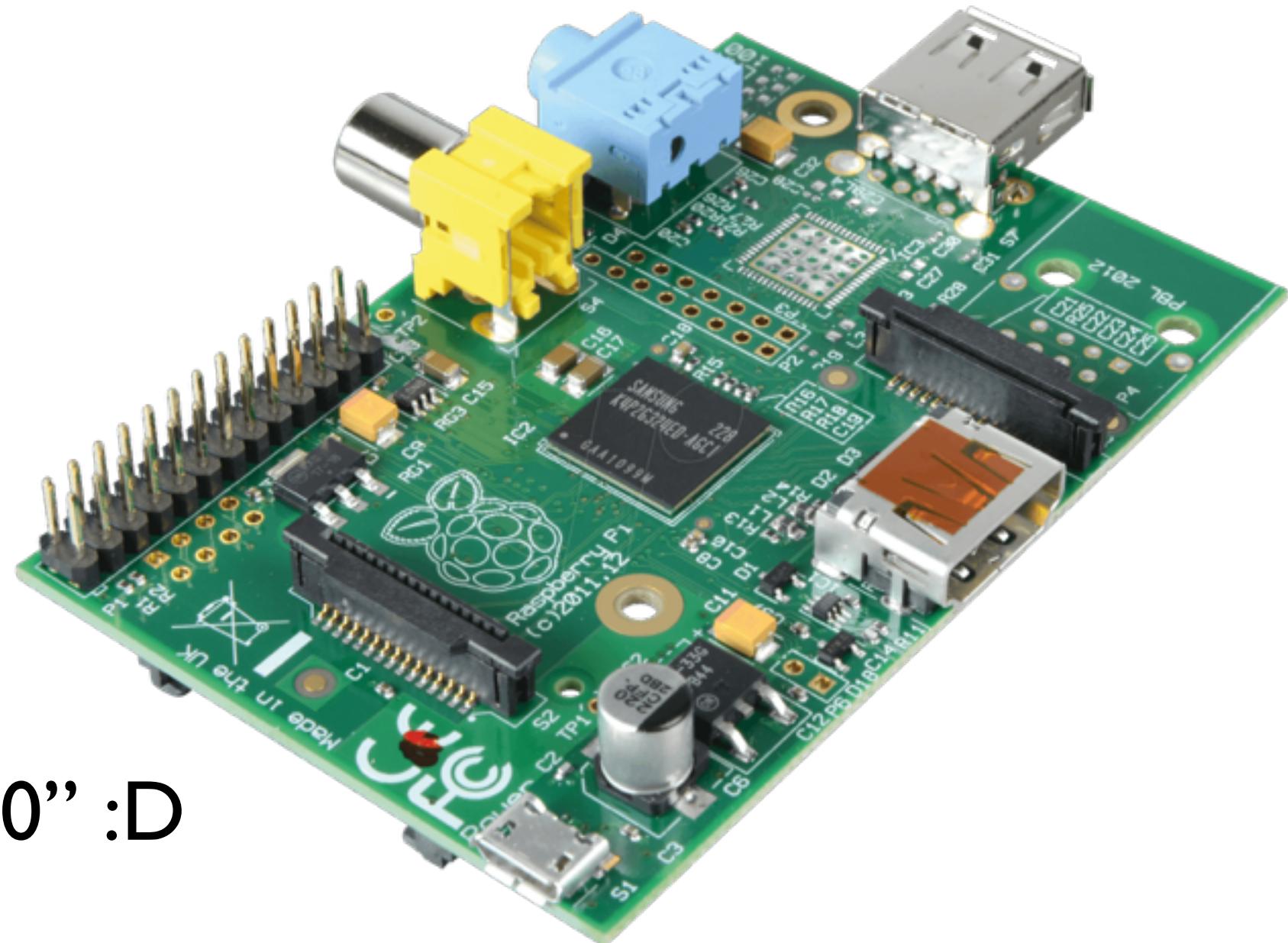
Mobile infrastructure



Raspberry Pi



- CPU 700 MHz
- 512 Mo de RAM
- 2 ports USB
- Carte SD
- Port ethernet
- “Un ordinateur comme en 2000” :D
- Le tout pour ~35\$





Mobile infrastructure



Demo



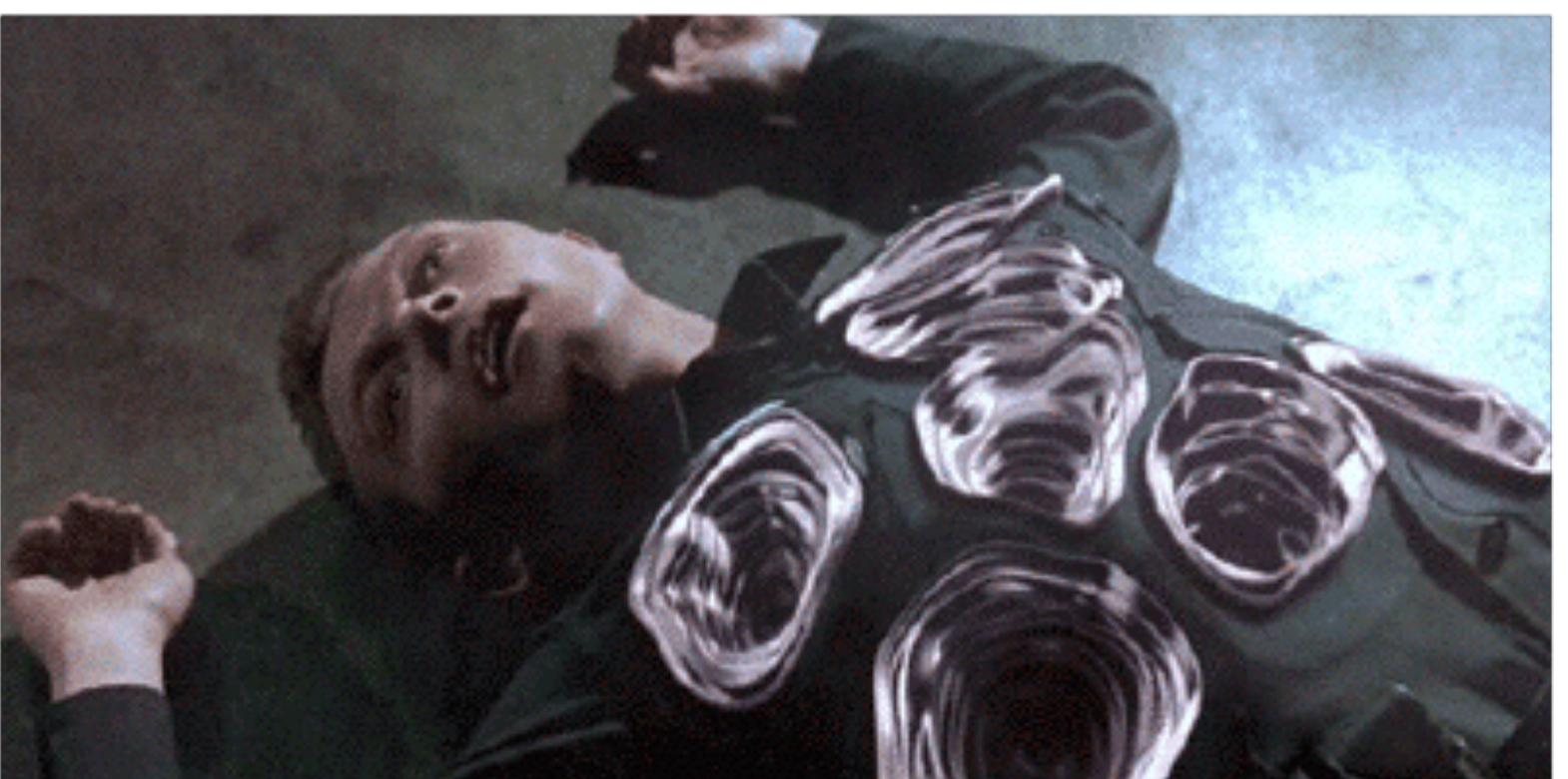


Résister à la charge



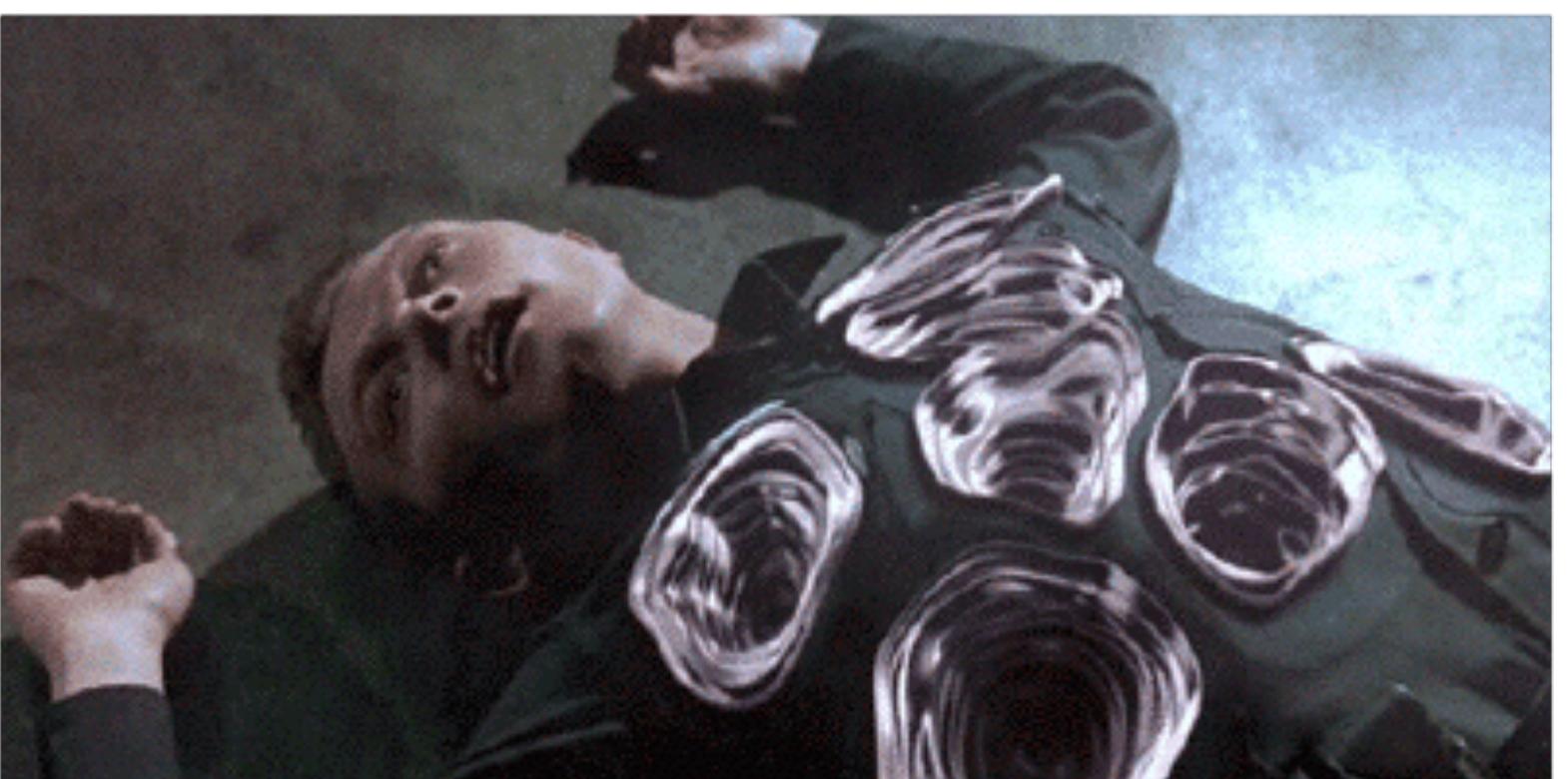


Résister aux pannes



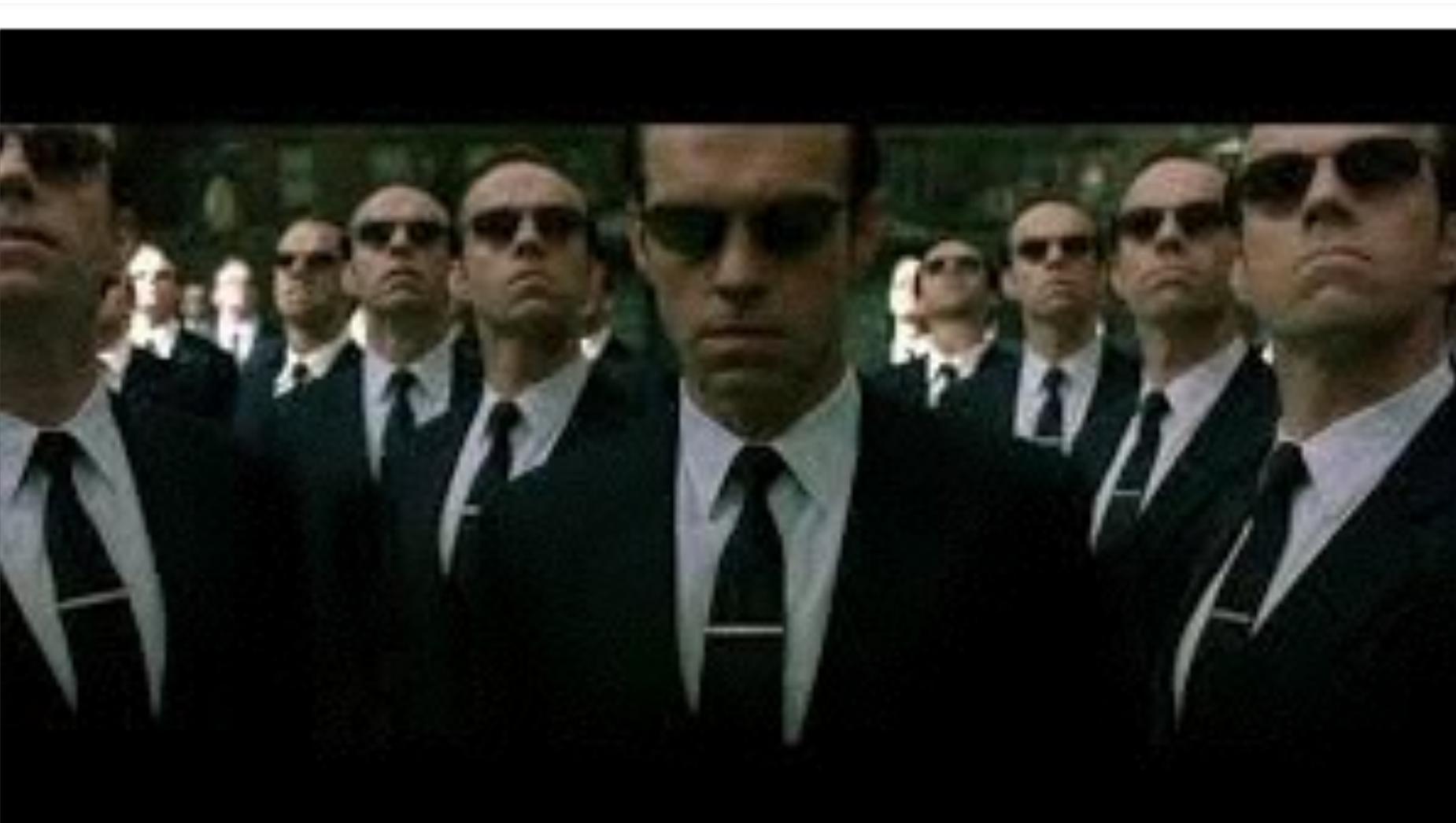


Résister aux pannes





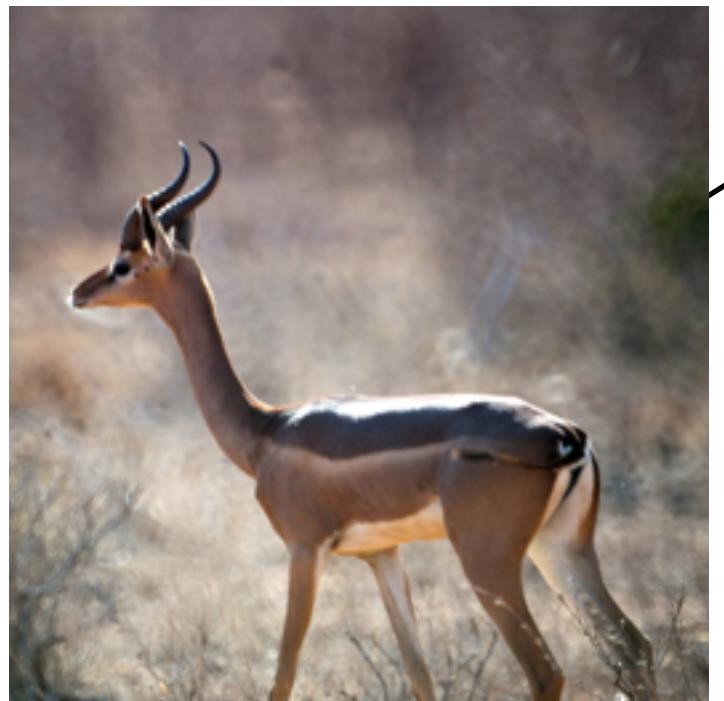
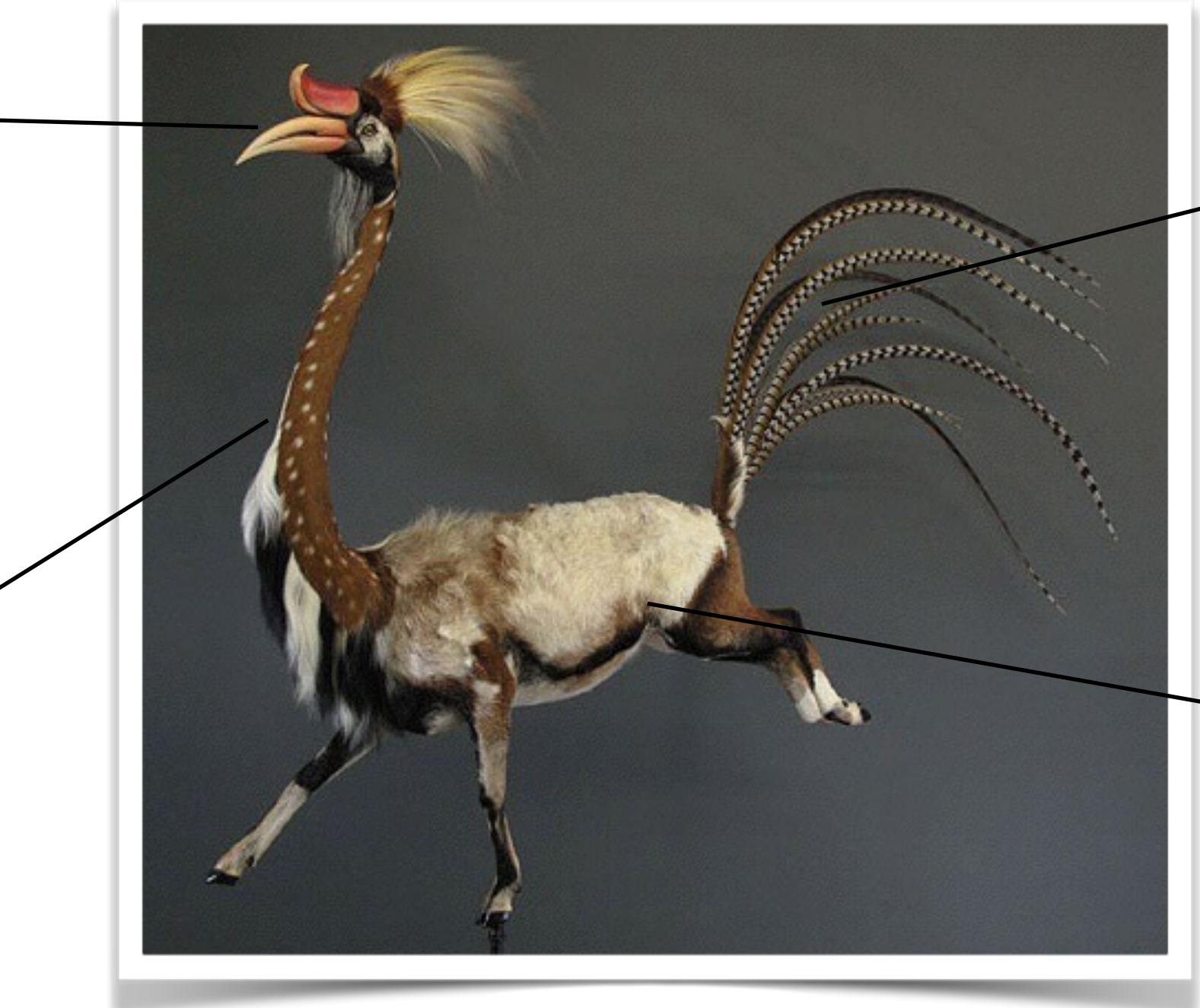
Absorber une hausse de trafic



Approche classique



Monolithique



Latence



SERLi



Bloquant



SERLi

Charge



SERLi

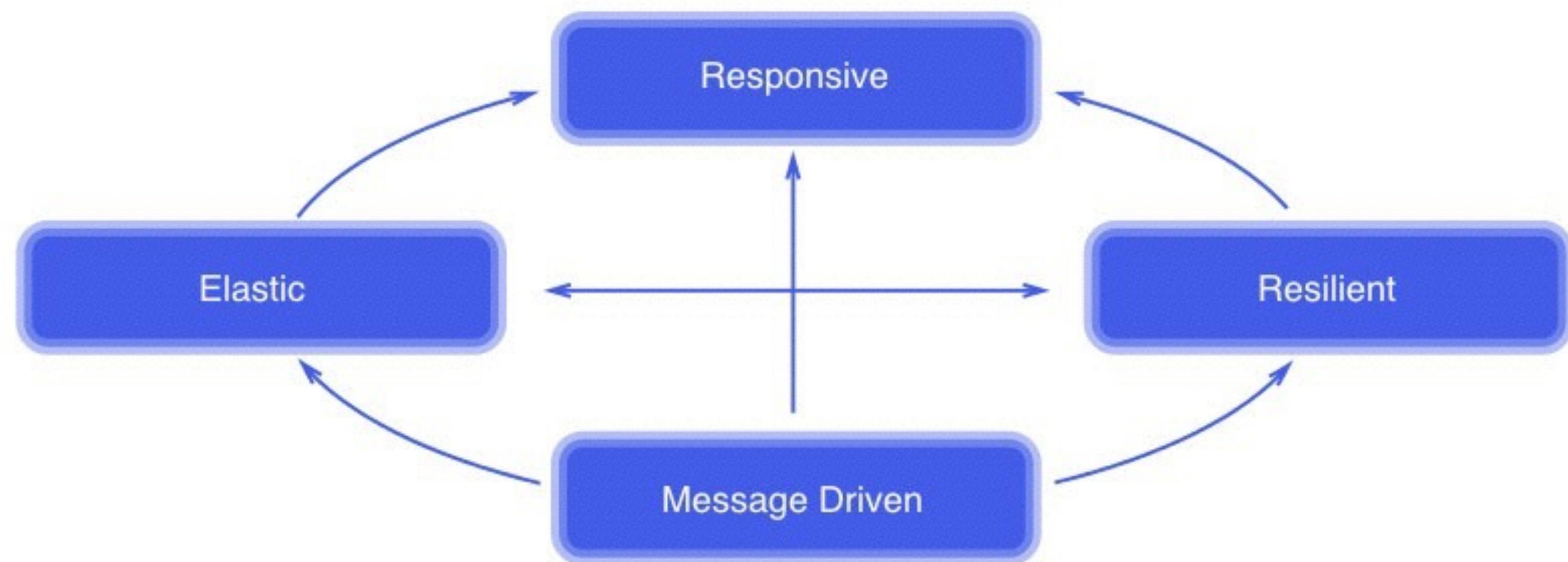
Scalabilité



Reactive Manifesto



Reactive Manifesto

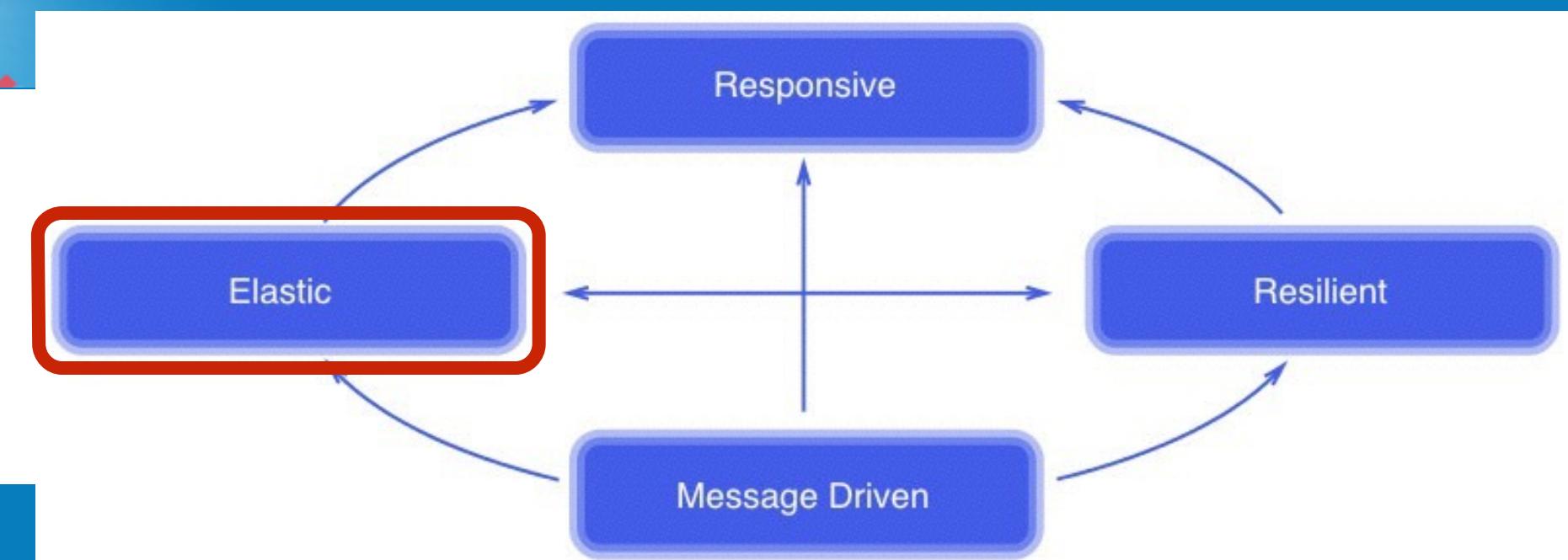




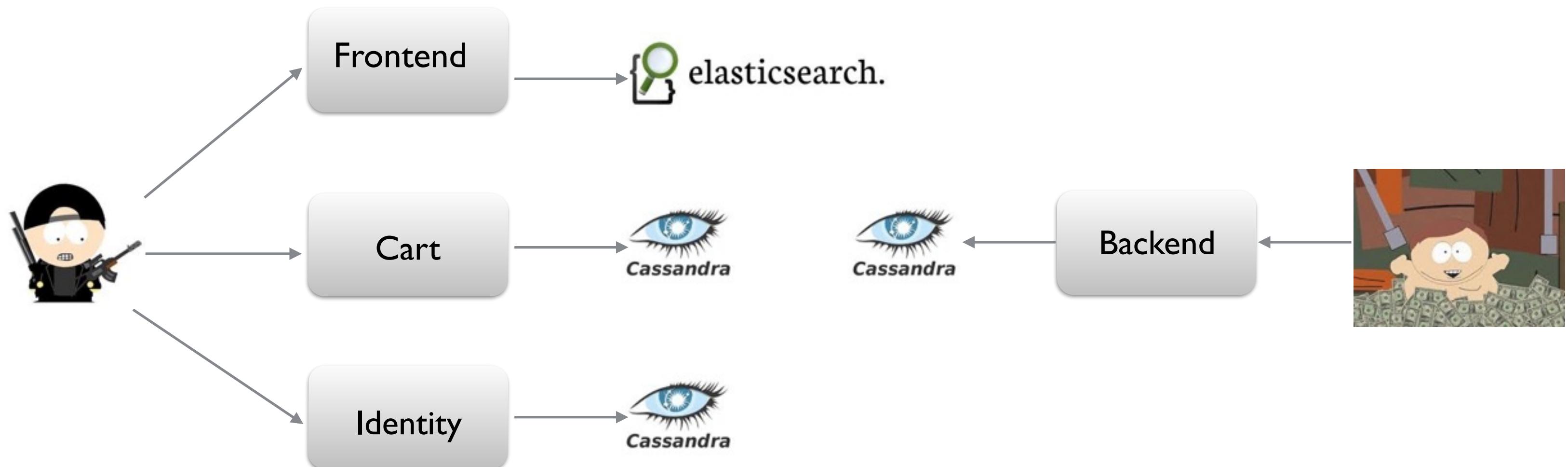
Reactive manifesto

- **react to event** : the event-driven nature enables the following qualities
- **react to load** : focus on scalability by avoiding contention on shared resources
- **react to failure** : build resilient systems with the ability to recover at all levels
- **react to user** : honor response time guarantees regardless of load

Scalable / React to load



Overview



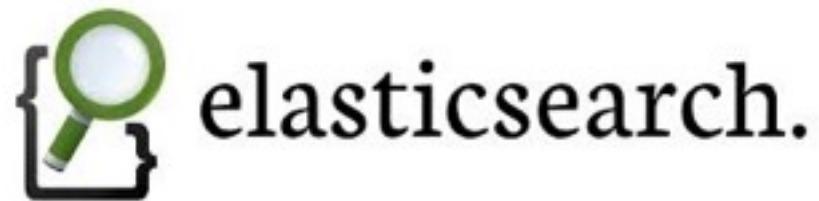


Datas



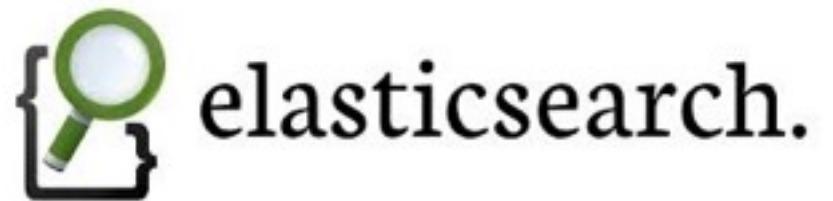


Datas





Datas





Mini / Micro Services

- Une application par domaine fonctionnel
 - store-frontend : présentation du contenu
 - store-identity : authentification / gestion de compte
 - store-cart : panier
 - store-backend : administration du site

Stateless



- Chaque application est stateless
 - aucune donnée n'est stockée dans l'application (pas de cache, pas de fichier ...)
- Chaque application peut être clonée



Stateless



- Chaque application est stateless
 - aucune donnée n'est stockée dans l'application (pas de cache, pas de fichier ...)
- Chaque application peut être clonée



Frontend



Amazing Store Hello test 0 items Logout

AMAZING ZOMBIE STORE TO SHOOT THEM ALL !!!

Search a product Search

Mitraillette années 30 canon court
Mitraillette années 30. S ...
3599.0 €
Add to cart

Mitraillette années 30
Mitraillette années 30. P ...
3599.0 €
Add to cart

Arbalète et accessoires
Magnifique arbalète pouli ...
200.99 €
Add to cart

- 100 % html
- Indexation par les moteurs de recherche
- stateless
- une url == un contenu



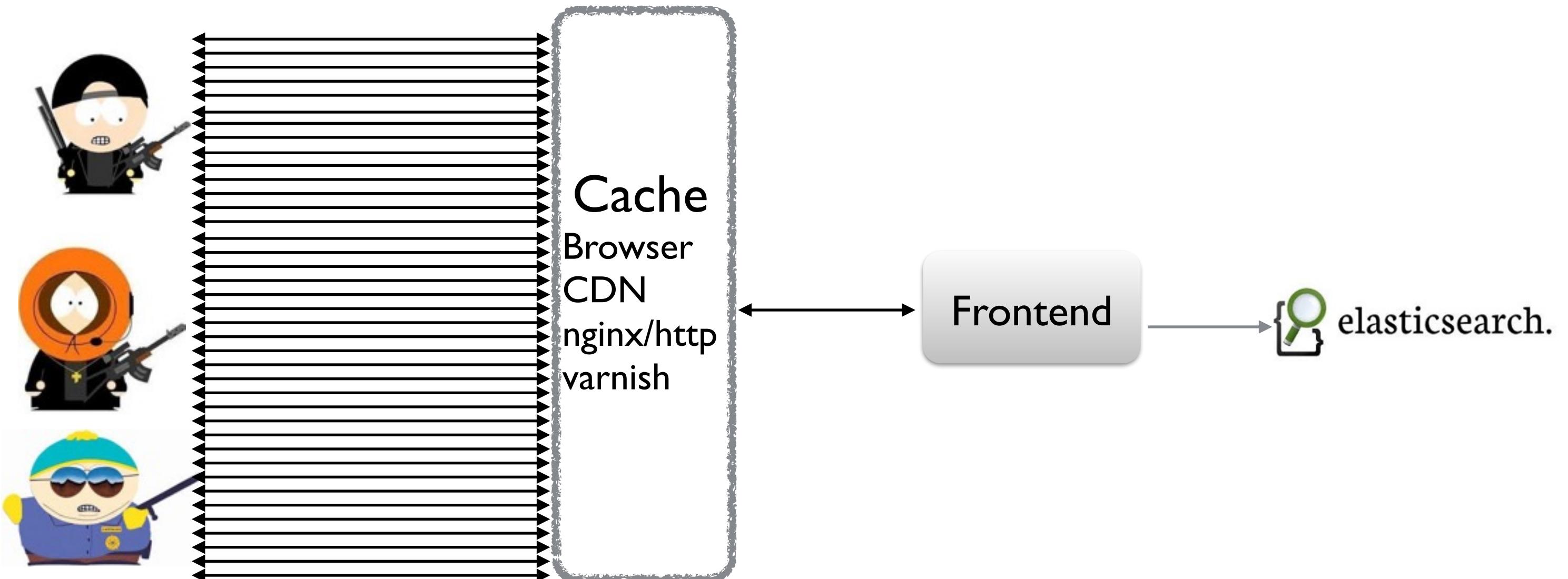
Limiter la charge



Limiter la charge



Cache





Optimisations

Base de données ?



elasticsearch.

Cache

+

recherche full text



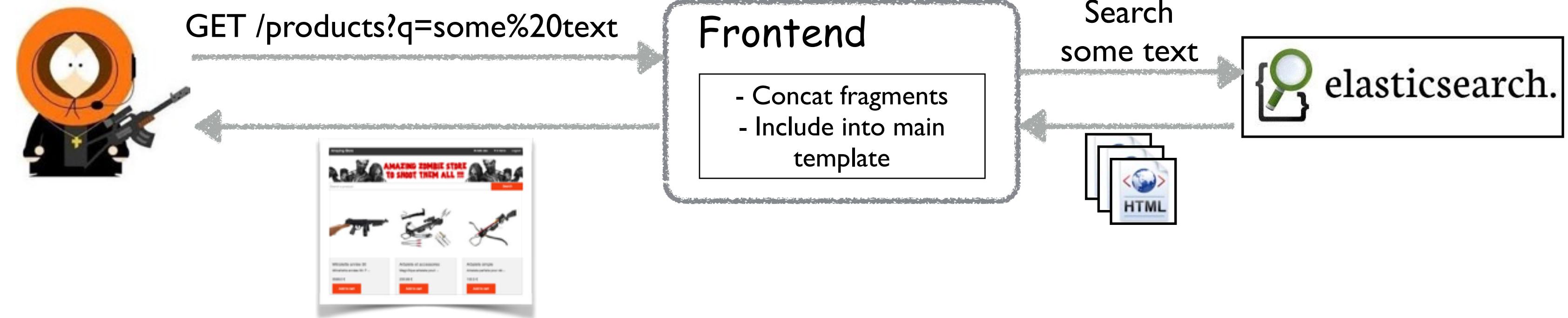
Modèle de données

```
{  
  id: "04abe480-2521-11e4-acde-f7b0d99b8321",  
  label: "Product number 1",  
  description: "A description ...",  
  image: "image.jpeg",  
  price: 1.5,  
  fragments: [{  
    type: "search",  
    html: " <div>...</div>"  
  }, {  
    type: "cart",  
    html: " <tr>...</tr>"  
  }]  
}
```

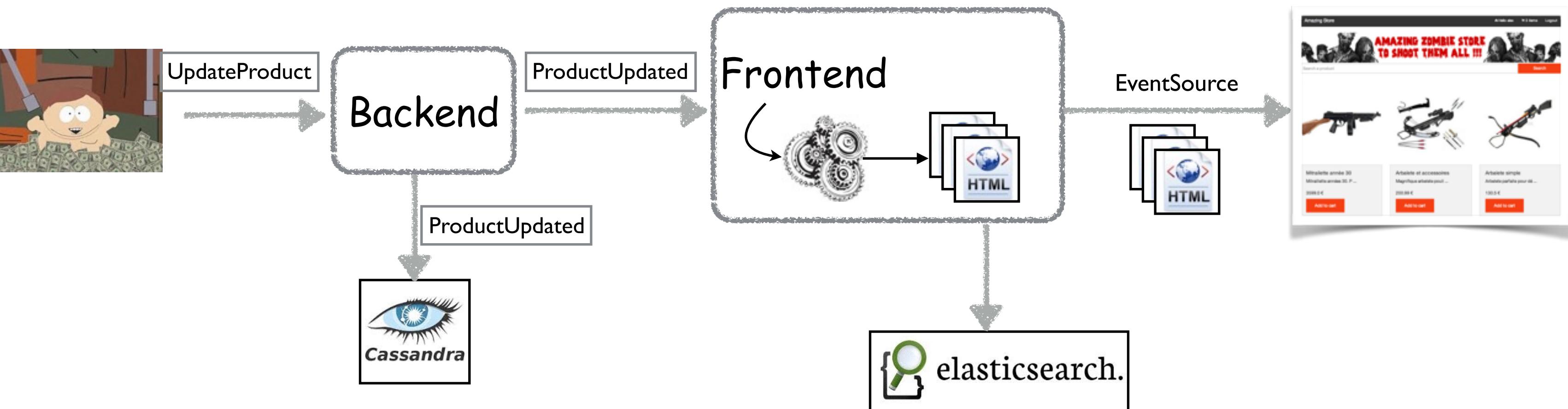
Données indexées pour la recherche

HTML pré-généré

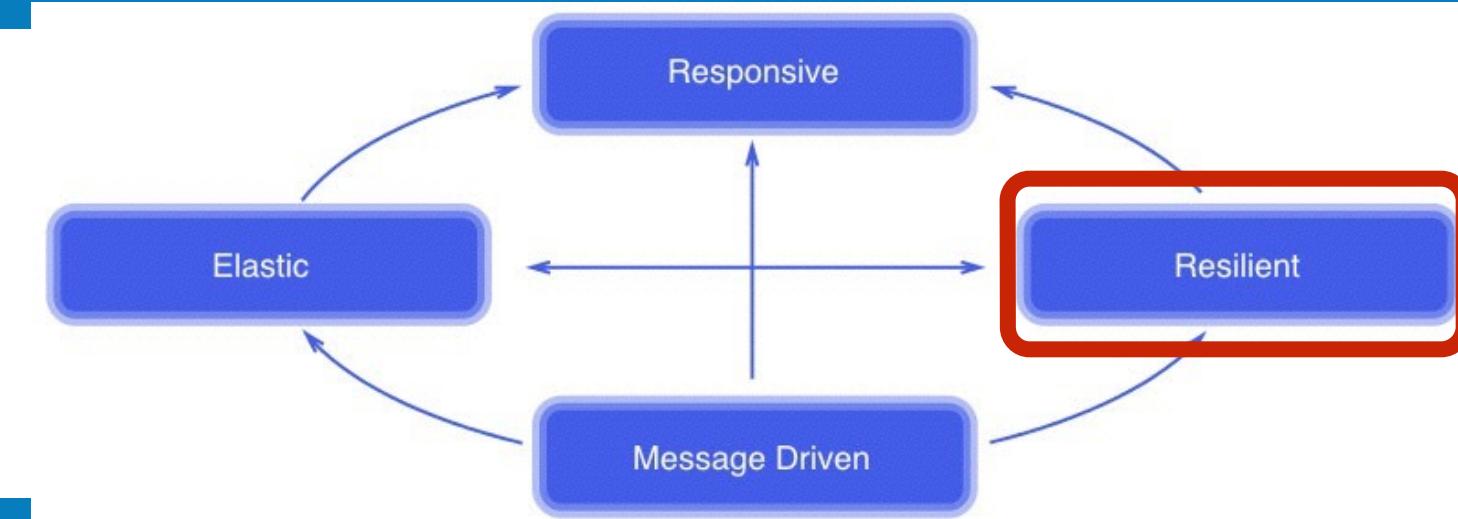
Recherche



Créer / mettre à jour



Resilient / react to failure

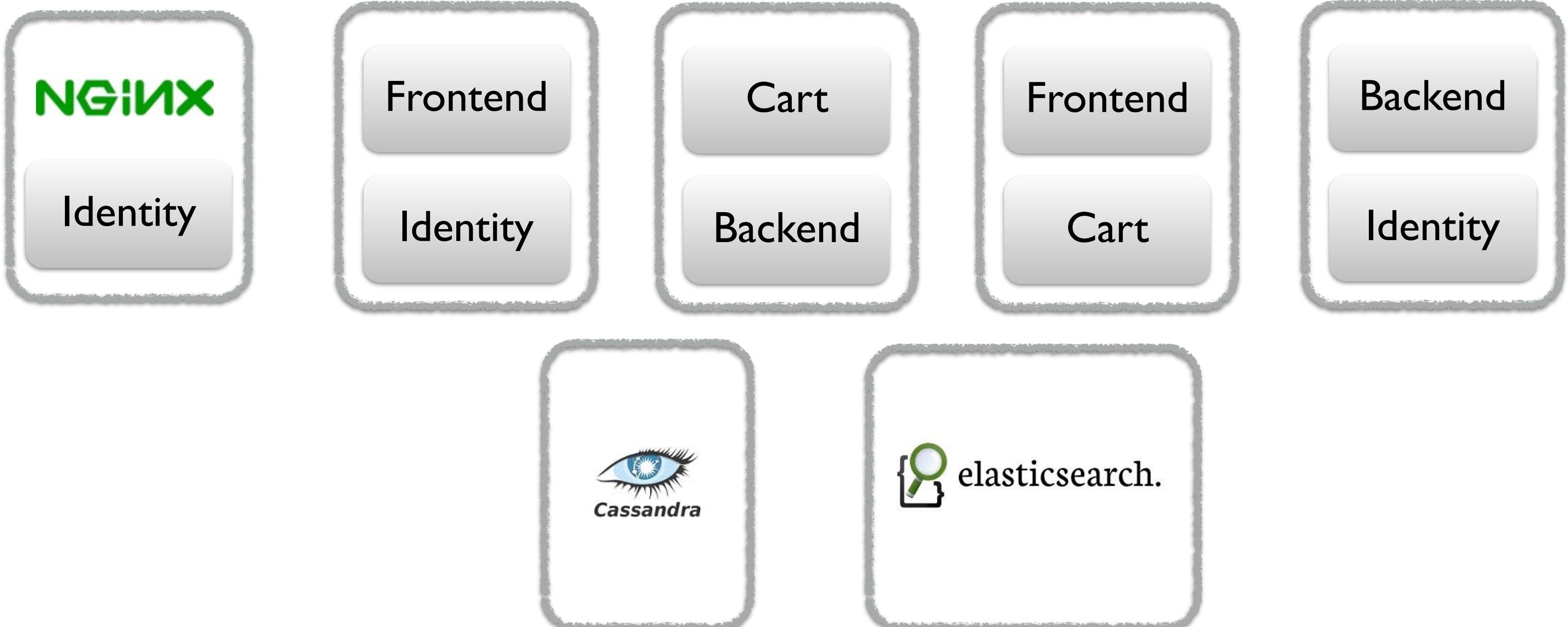




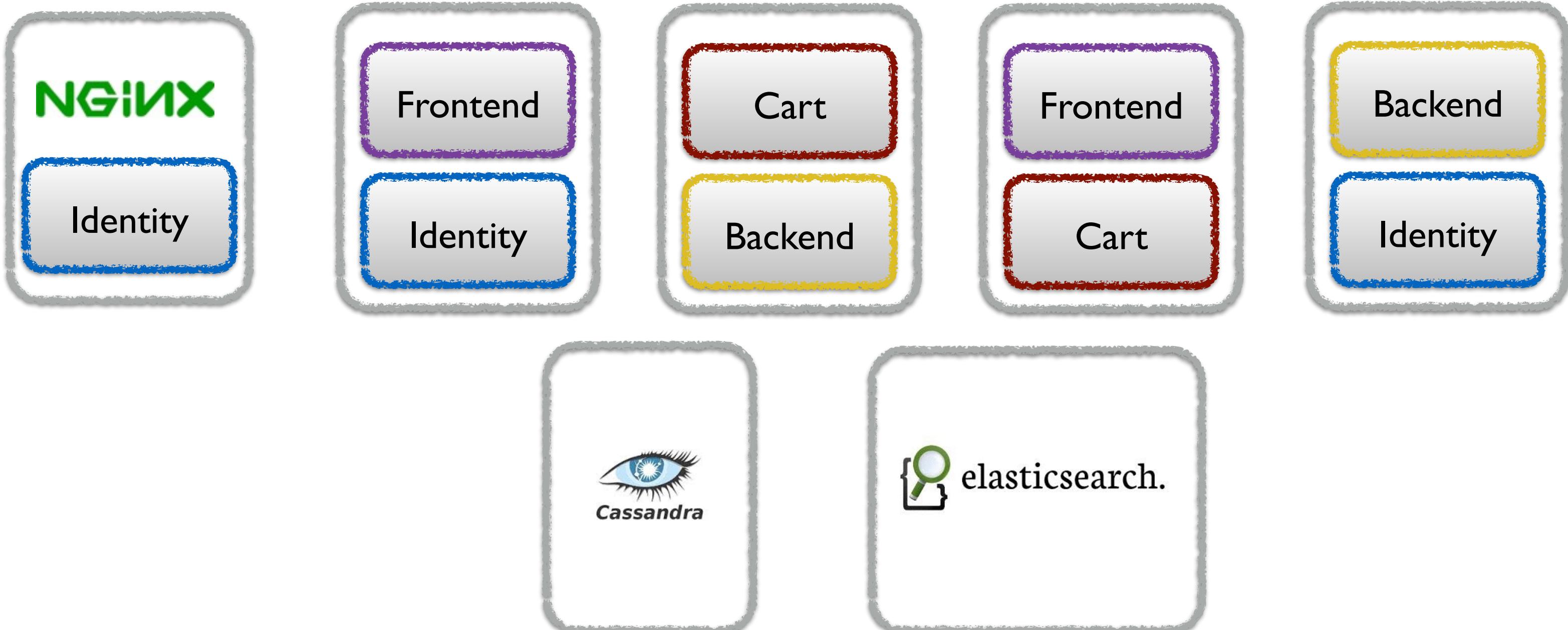
SERLi



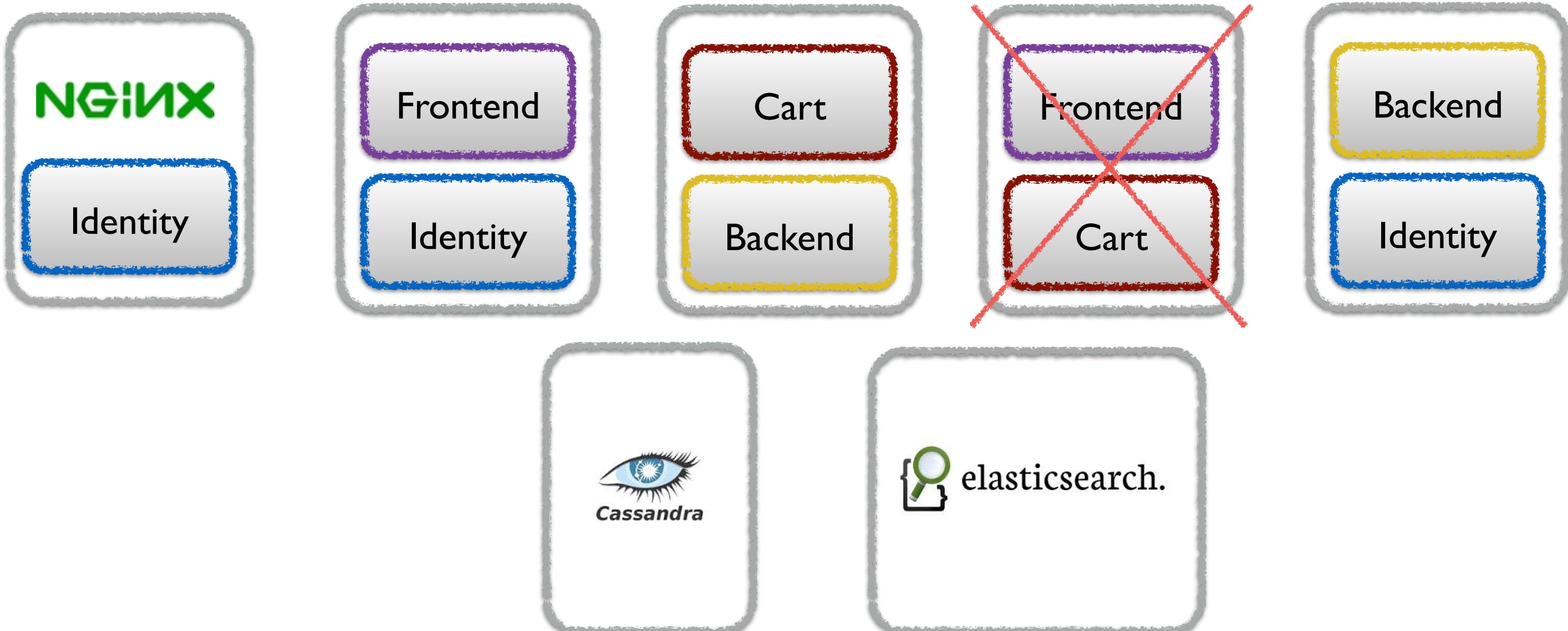
Infrastructure



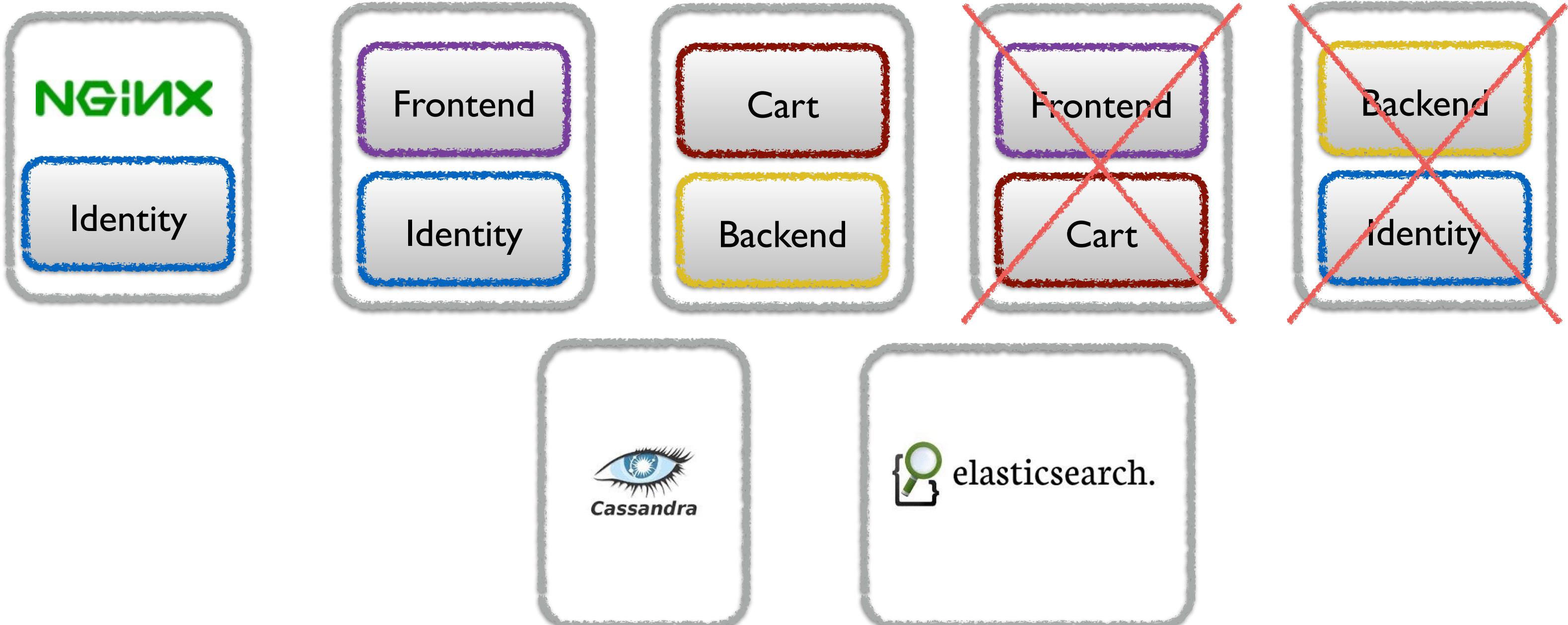
Infrastructure



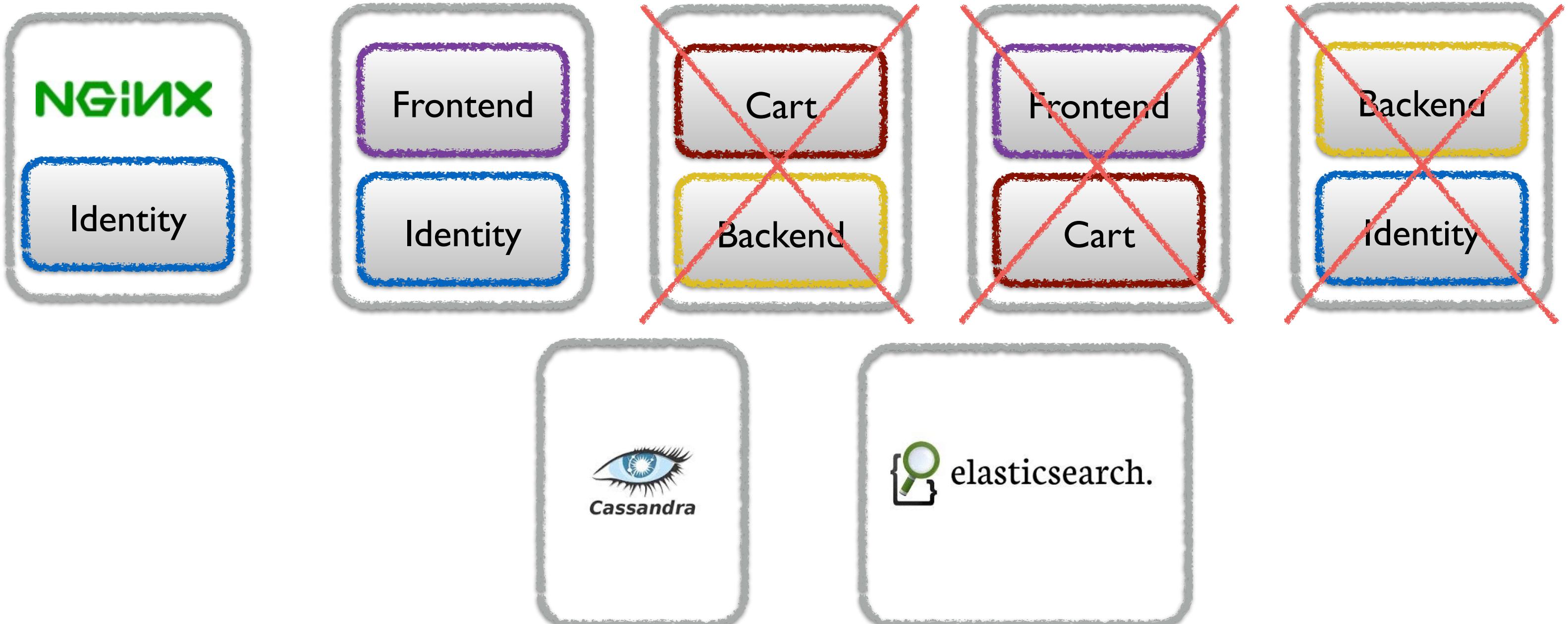
Infrastructure



Infrastructure



Infrastructure

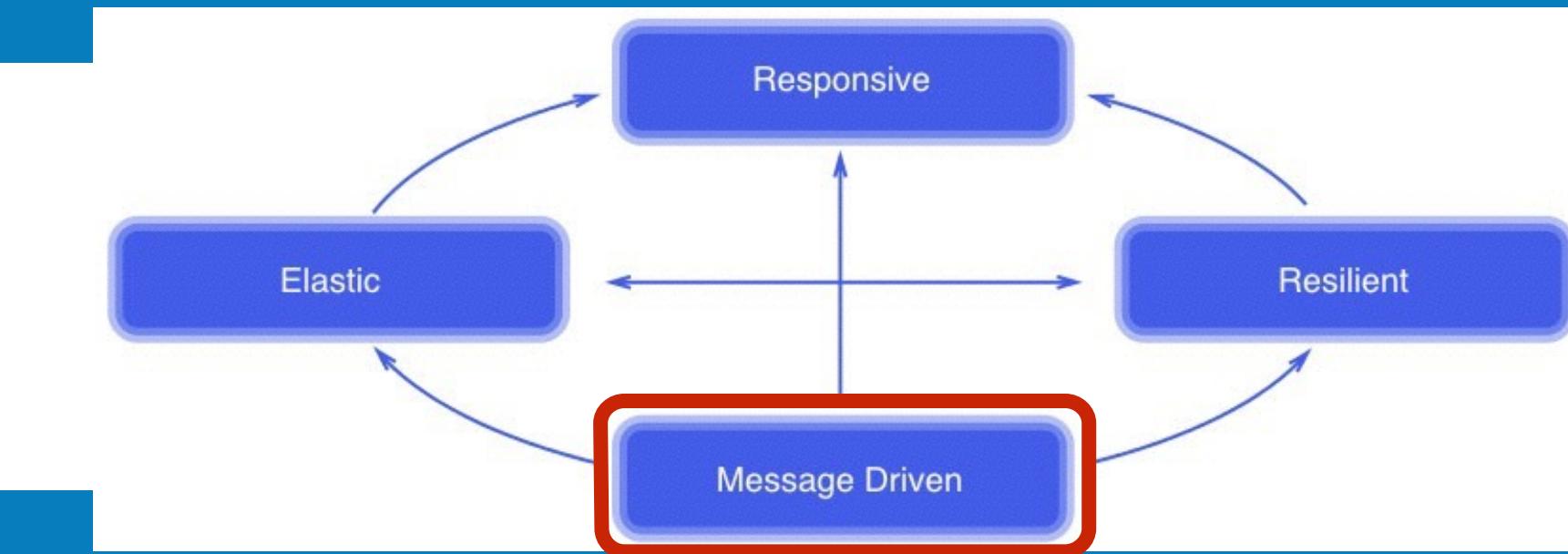


Demo

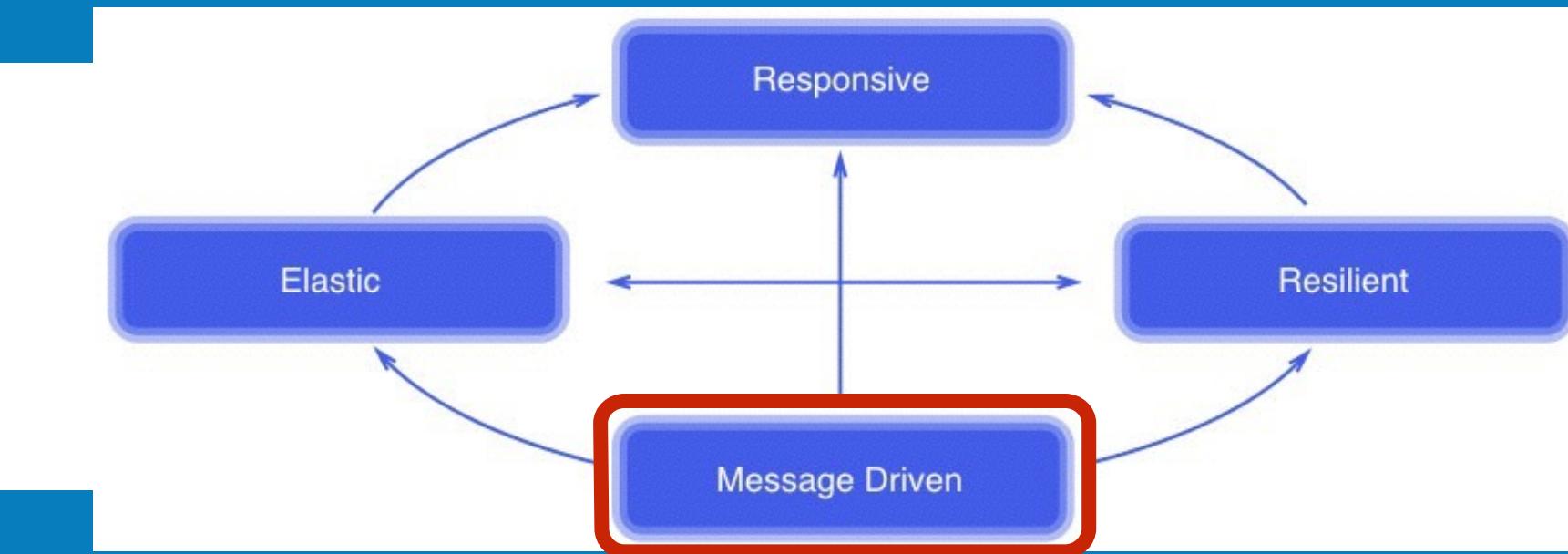
Let It Crash !!!



Event-driven / react to event



Event-driven / react to event



Akka ?



SERL.

MAKE GIFS AT GFSOUP.COM

Akka ?



SERL.

MAKE GIFS AT GFSOUP.COM



Akka ??



SERLi

- Akka
 - Modèle acteur
 - Un acteur = Une entité statefull
 - Communication entre acteurs par messages (même à distance)
 - Un acteur peut créer/détruire des enfants
 - Un acteur peut surveiller d'autres acteurs
 - Plus de problèmes de concurrence, asynchrone par nature
 - Résistant aux pannes
 - Java or Scala





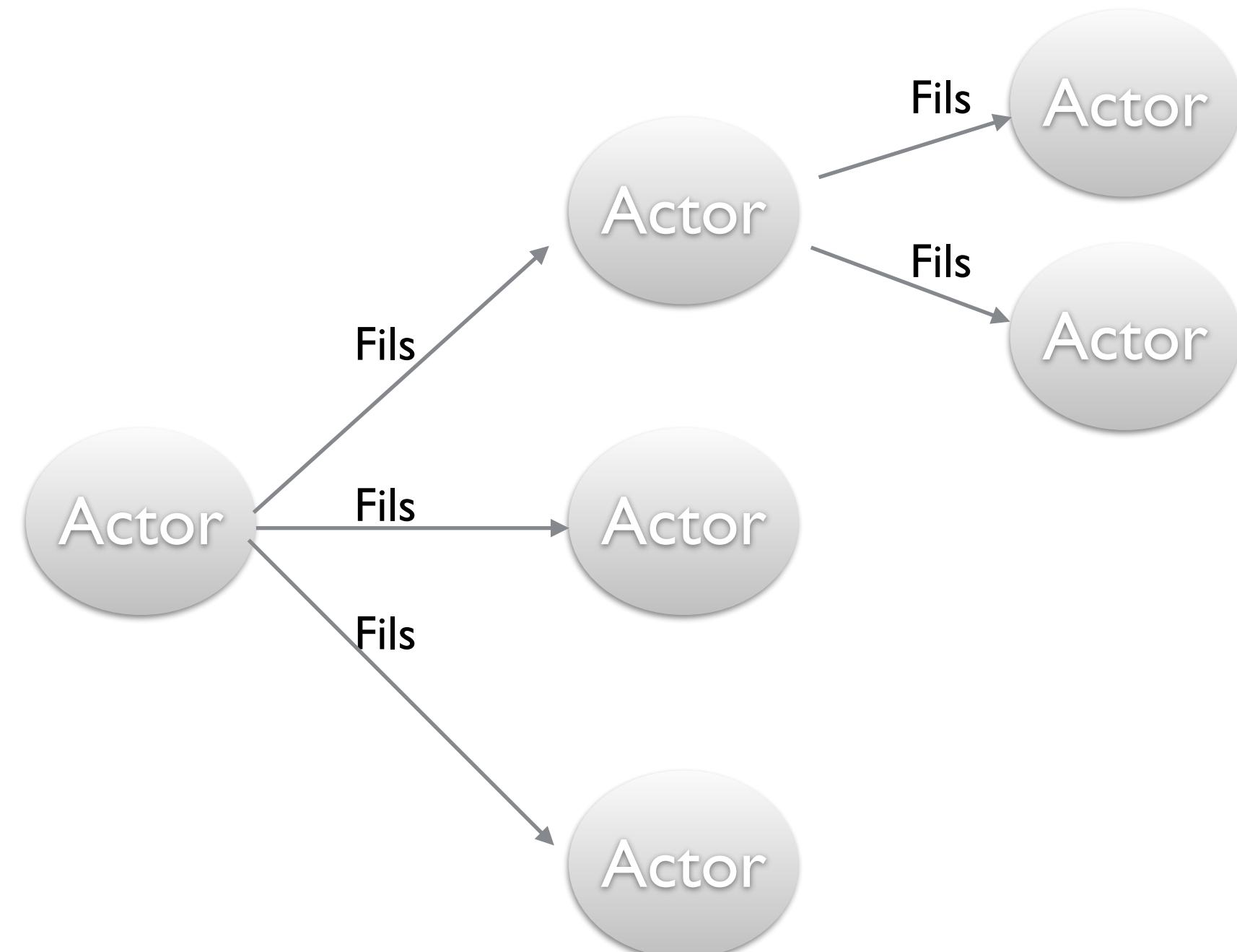
Akka

Actor

SERLi

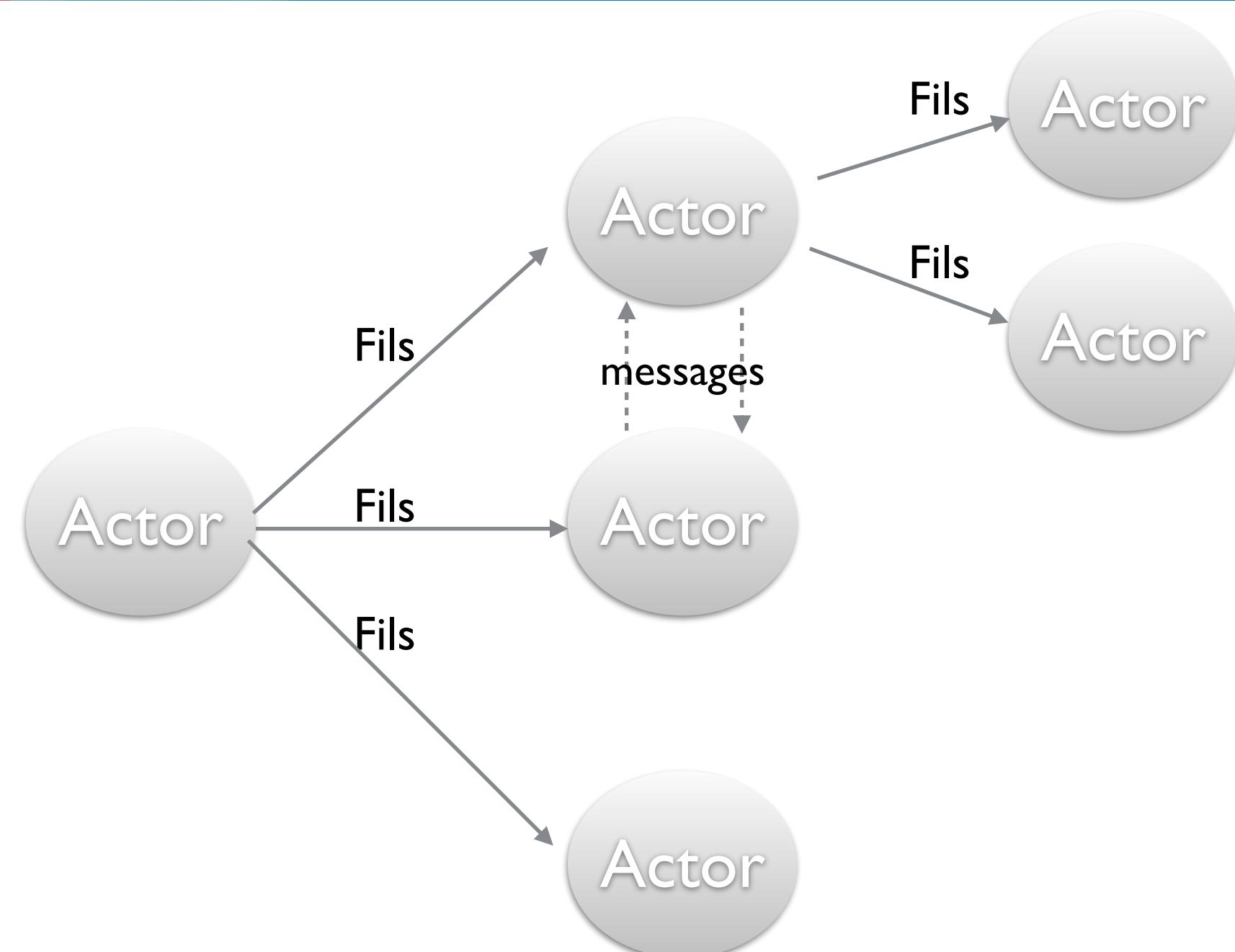


Akka



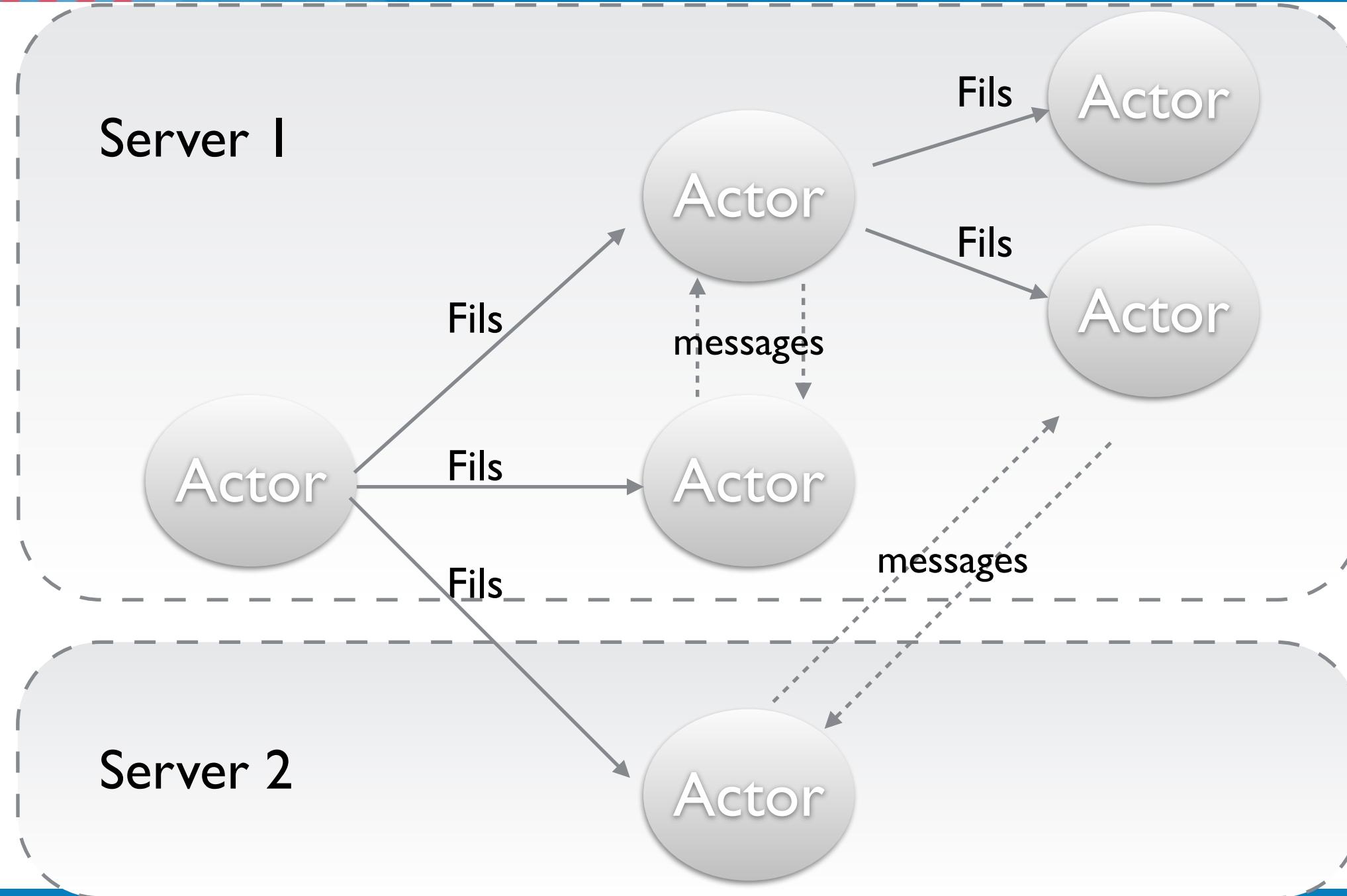


Akka



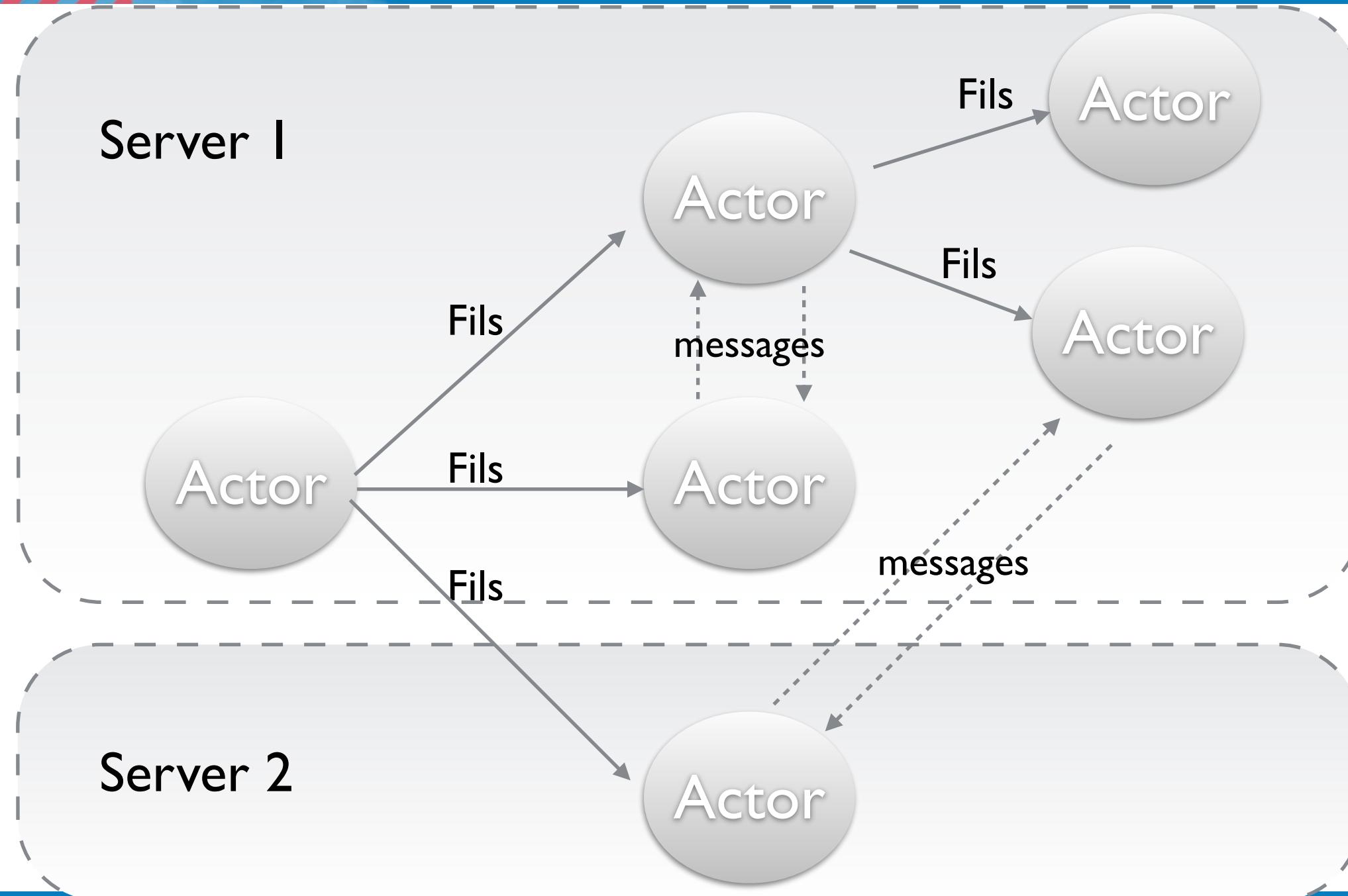


Akka





Akka





Akka messages

```
import akka.actor.{Props, ActorSystem, ActorLogging, Actor}

case class Greeting(who: String)

class GreetingActor extends Actor with ActorLogging {
  def receive = {
    case Greeting(who) => log.info("Hello " + who)
  }
}

val system = ActorSystem("MySystem")
val greeter = system.actorOf(Props[GreetingActor], name = "greeter")
greeter ! Greeting("Charlie Parker")
```



ASP.NET



Struts²

SERLi



Play 2

- Framework web
- java or scala
- Support pour les websocket et le server sent event
- Asynchrone
- pré-requis pour une application orientée événements





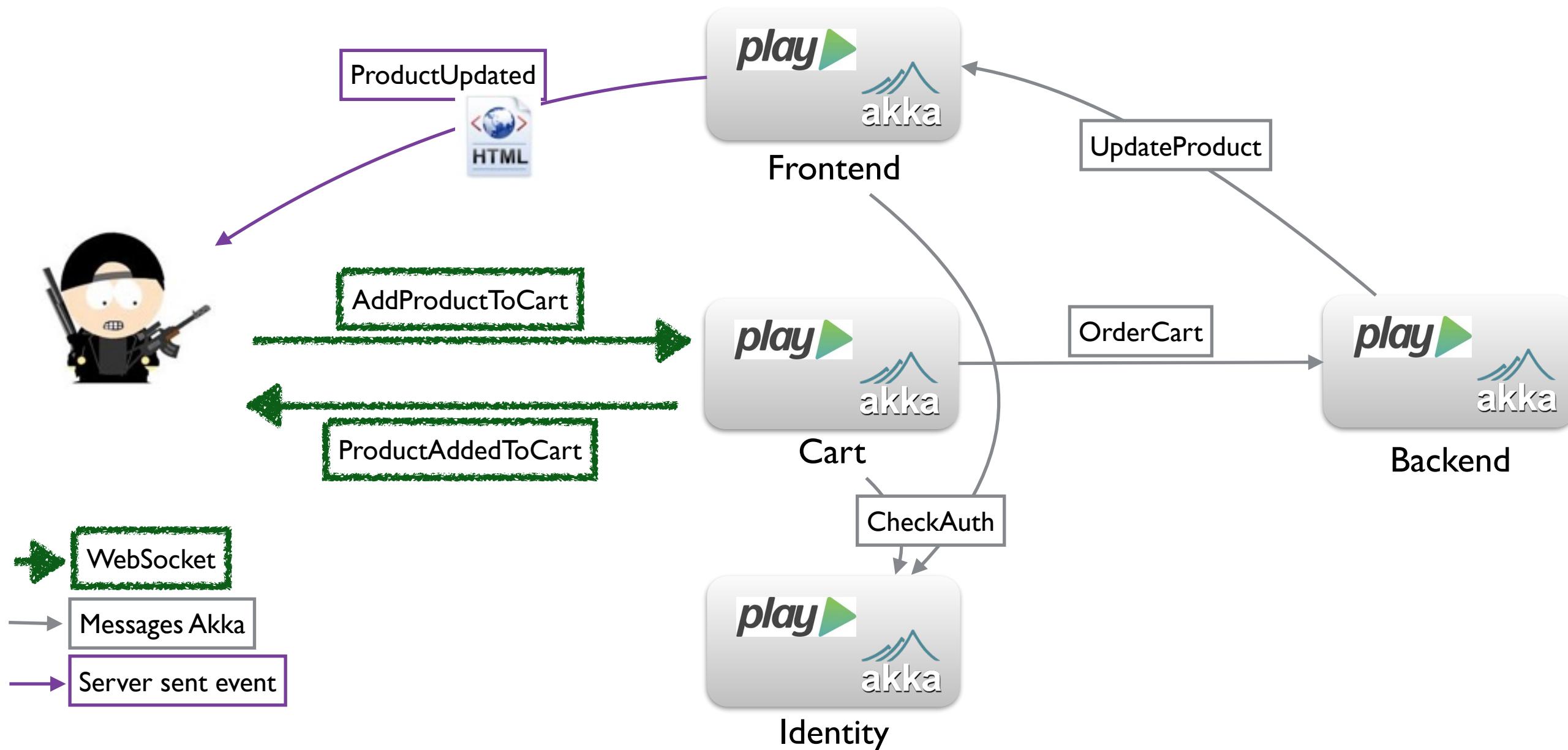
Play async

```
case class ListProductsQuery()

class ProductView extends Actor {
  override def receive: Receive = {
    case ListProductsQuery() => models.Product.list() pipeTo sender()
  }
}

class ProductsController extends Controller {
  private def listProducts(): Future[List[models.Product]] = {
    (Actors.productView() ? ListProductsQuery()).mapTo[List[models.Product]]
  }
  def index() = Action.async {
    listProducts().map(products => Ok(views.html.index(products)))
  }
}
```

Messages

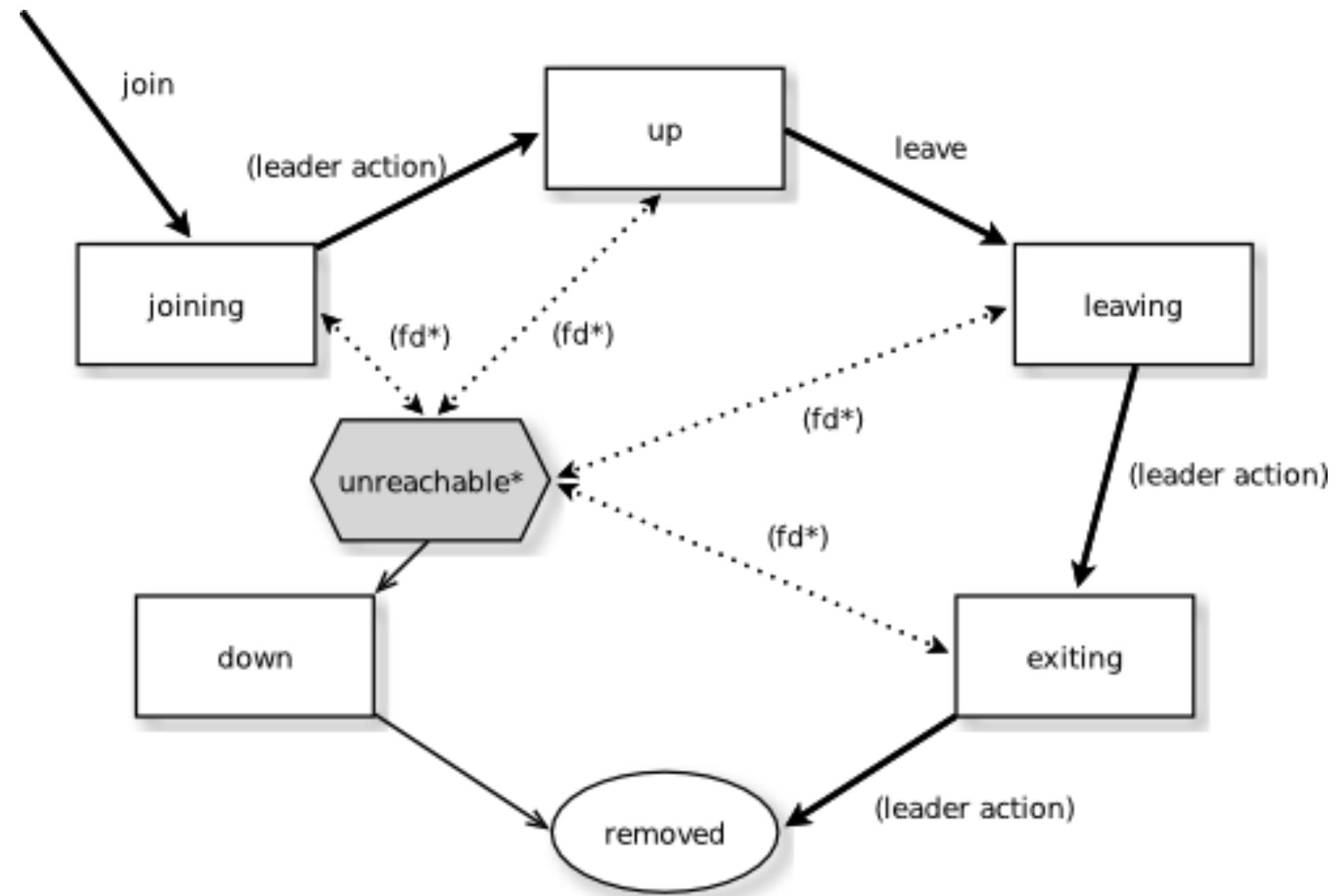




Cluster

- Utilisation de Akka-cluster
- Librairie permettant de former un cluster de systèmes d'acteurs
 - simple service d'adhésion
 - tolérant aux pannes
 - décentralisé (P2P, gossip)
 - pas de SPOF
 - pas de SPOB
 - détection des pannes

Akka cluster



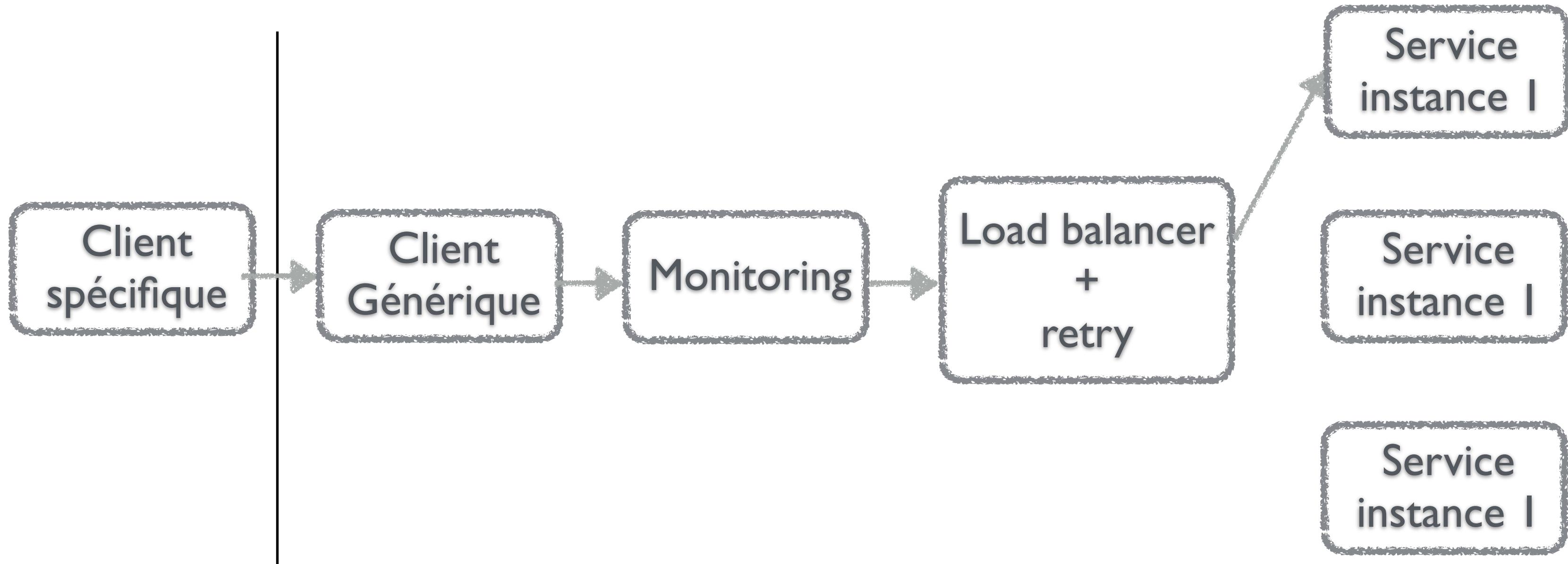


Cluster services

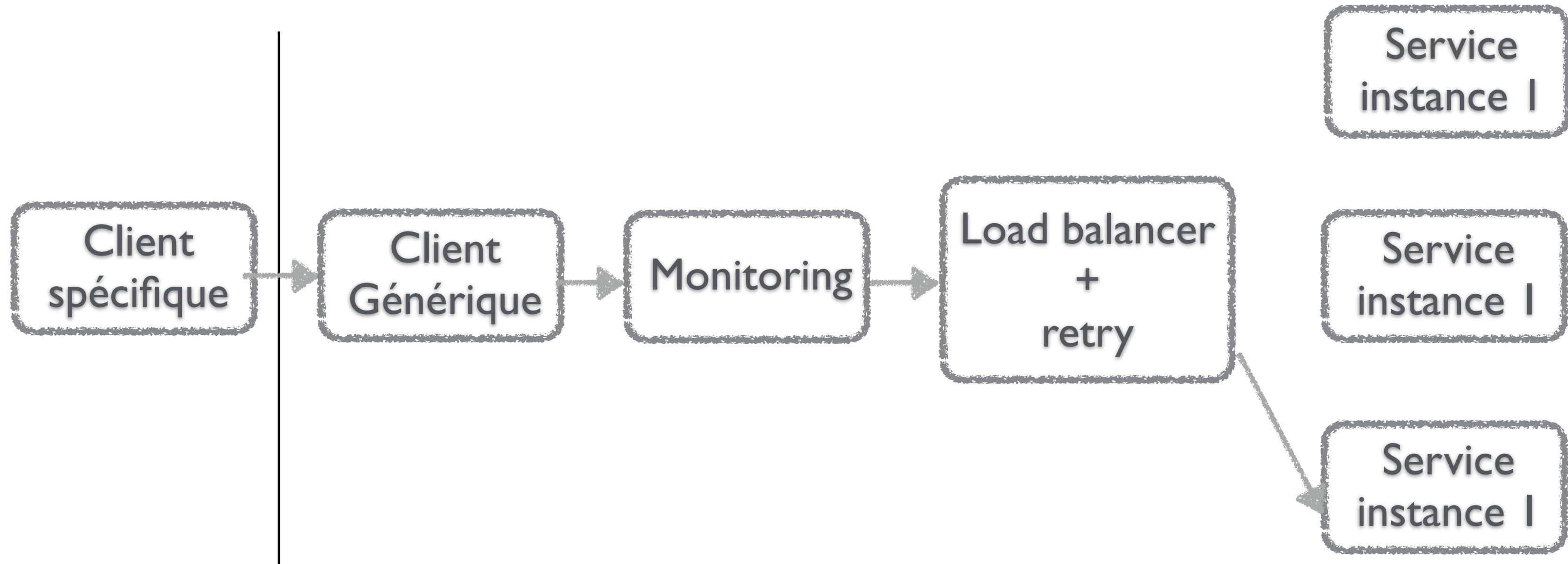
- Librairie de découverte de services distribués
- Exposition descripteurs de services (URL, protocole, version, nom)
- Repose sur les memberships du cluster Akka
- Clients de services
 - HTTP, Akka, Thrift, Memcached ...
- Petits plus
 - Monitoring
 - Load balancing (pas très intelligent)
 - Retry with exponential back off



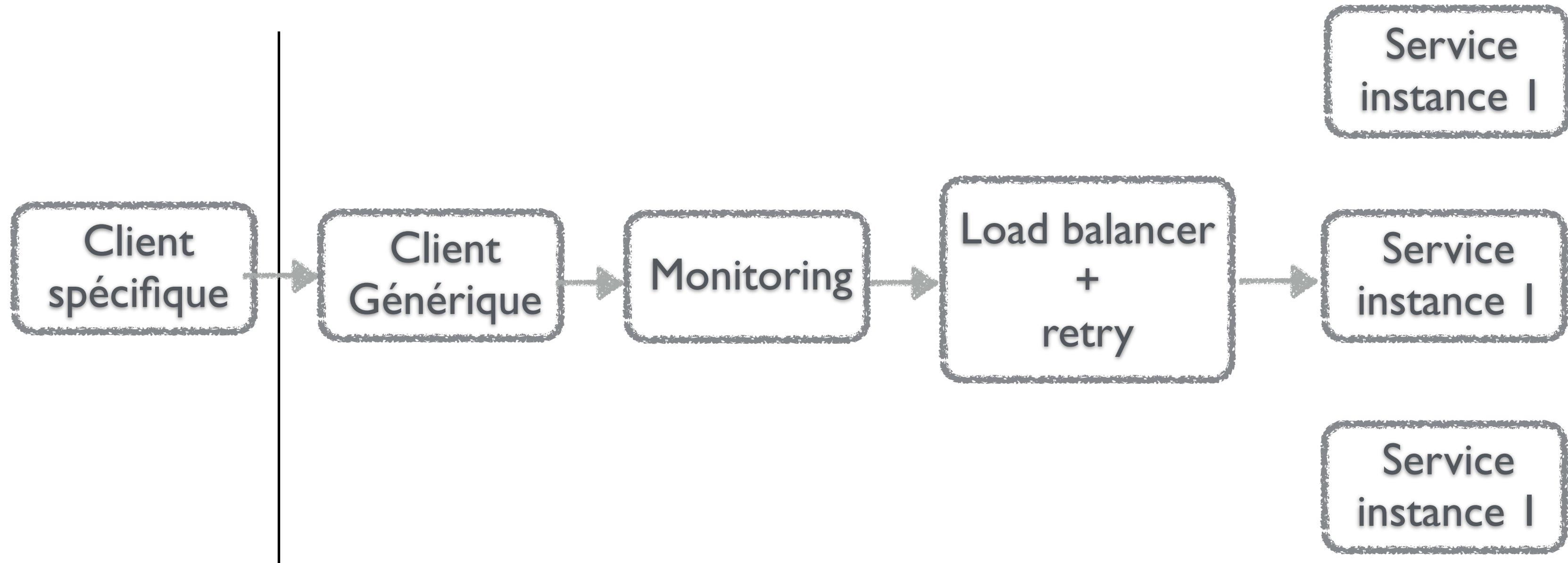
Cluster services



Cluster services



Cluster services



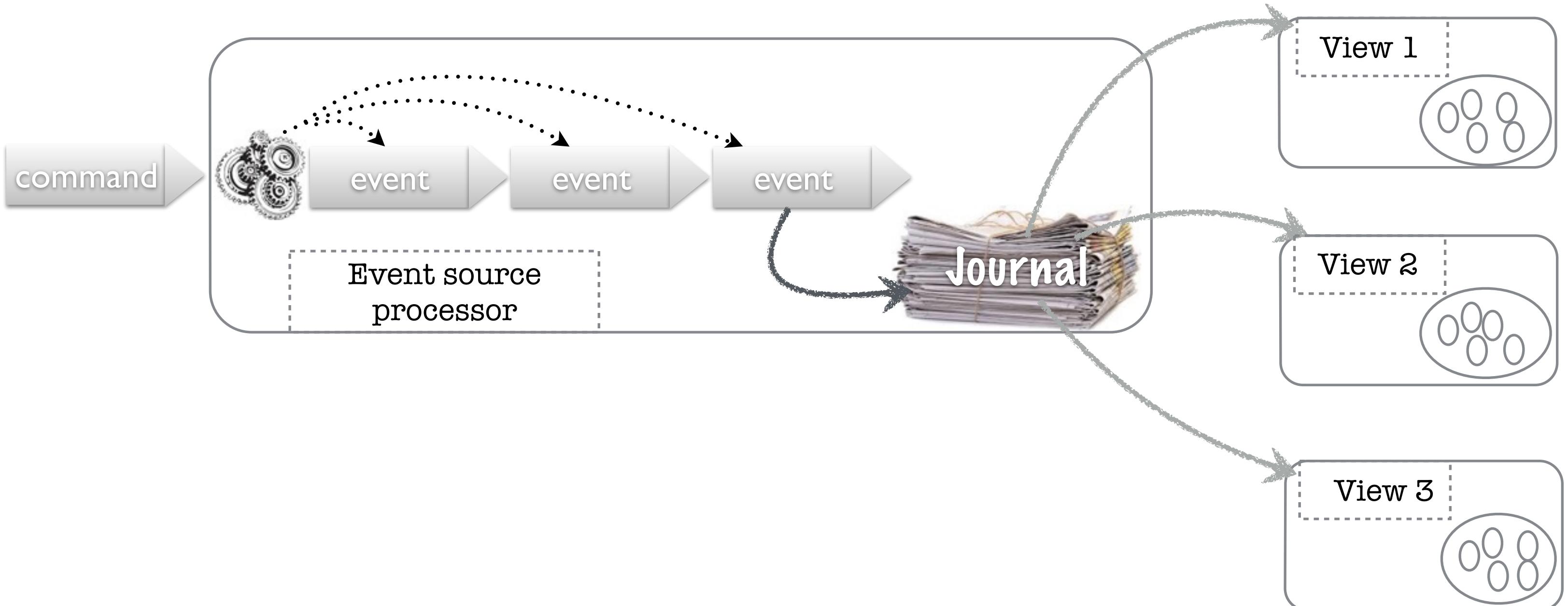


CQRS & EventSourcing

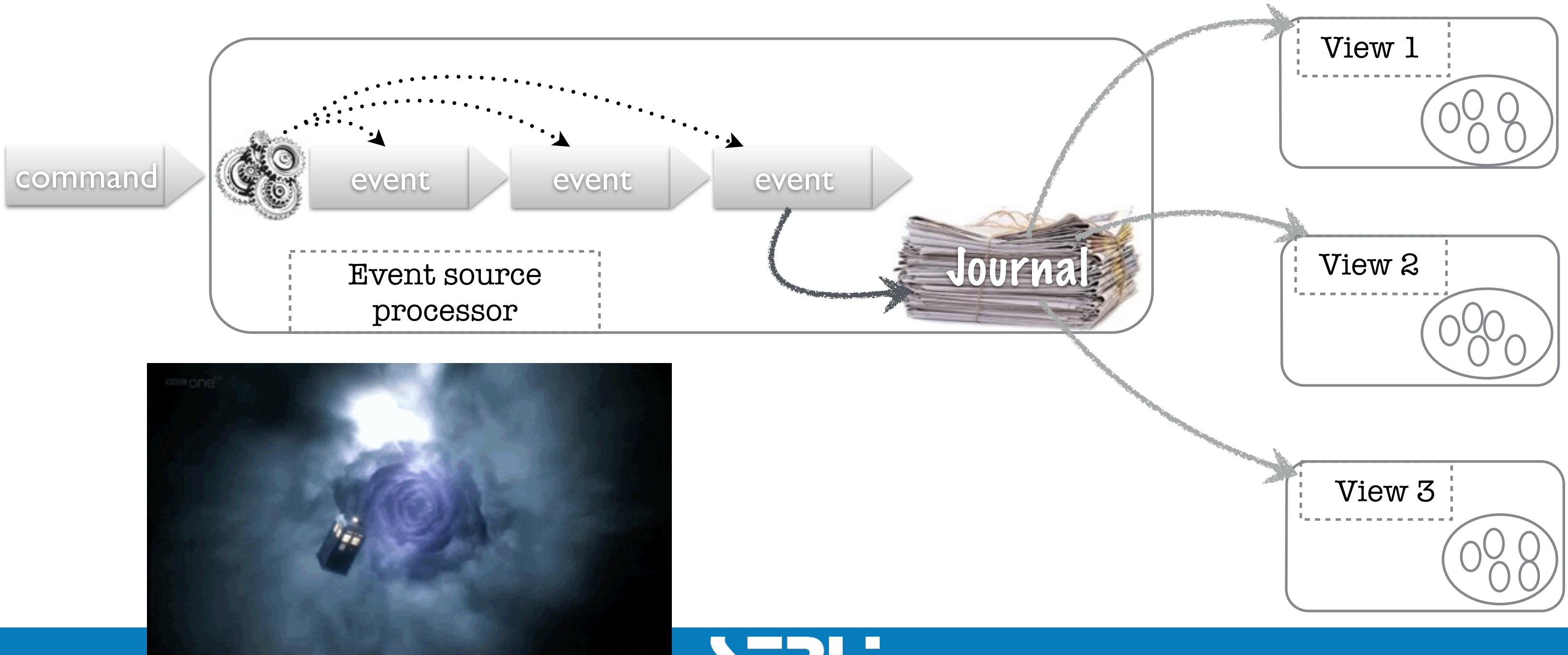
- Command Query Responsibility Segregation
 - Command : Enregistrement
 - Query : Lecture
 - 2 modèles distincts
 - Séparation des services
- Event sourcing
- Stockage des événements



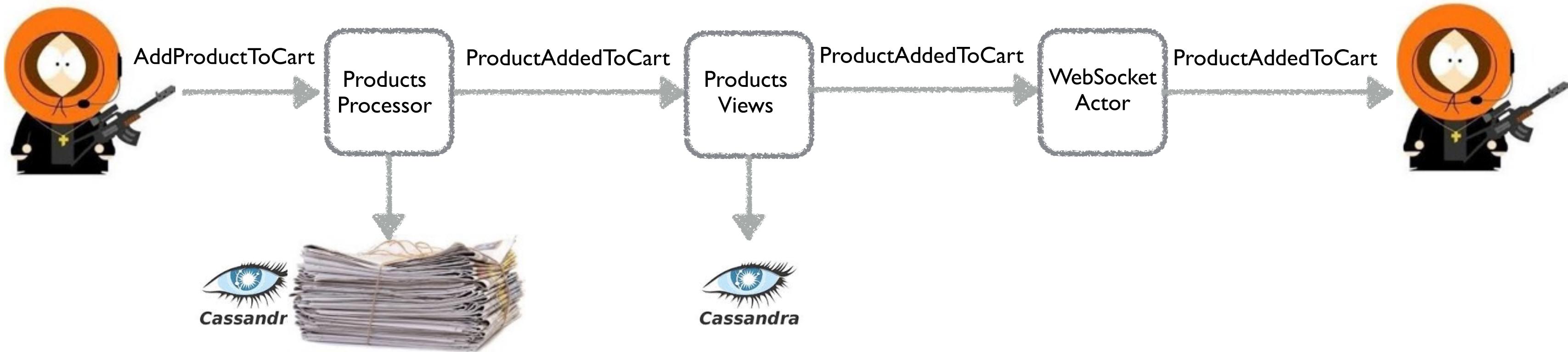
Persistence



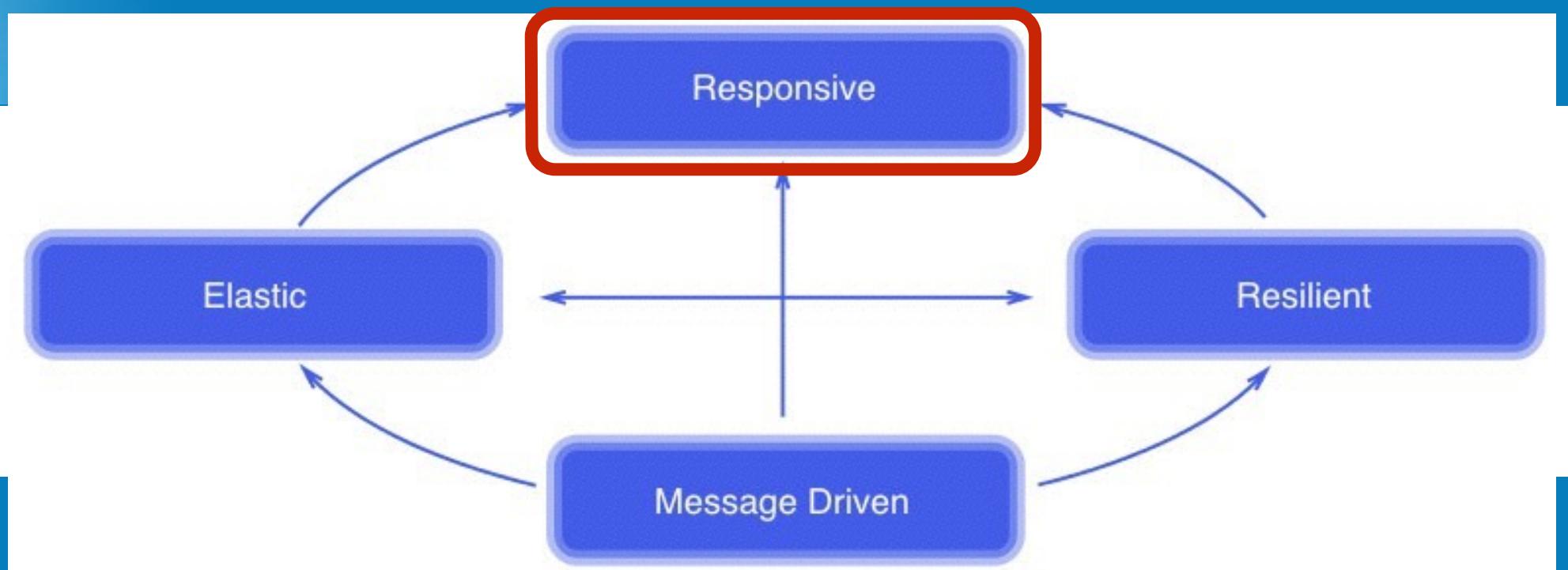
Persistence



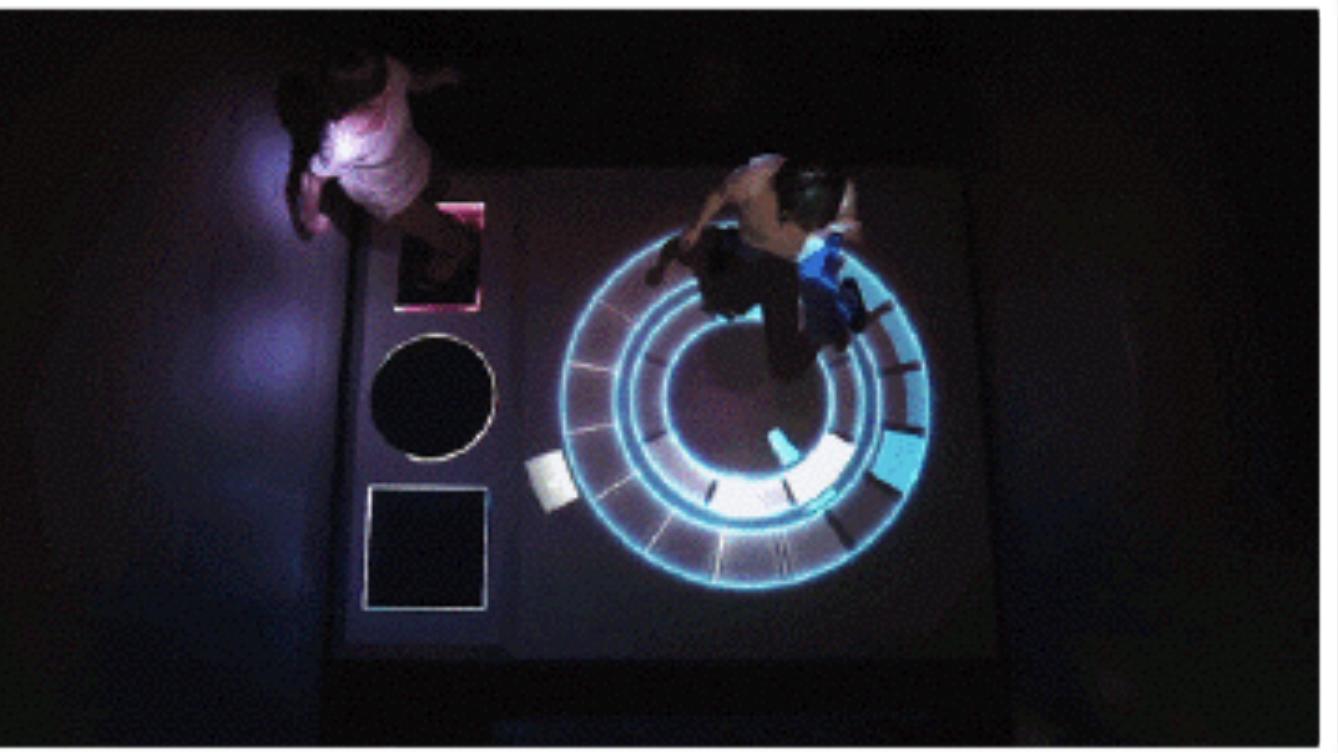
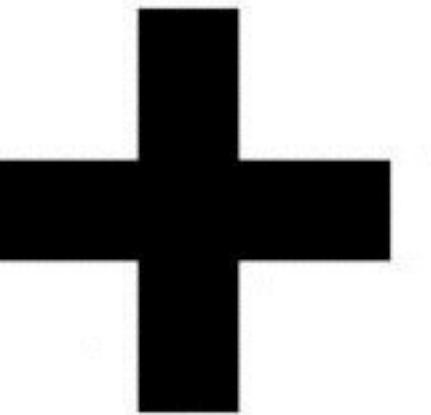
Exemple



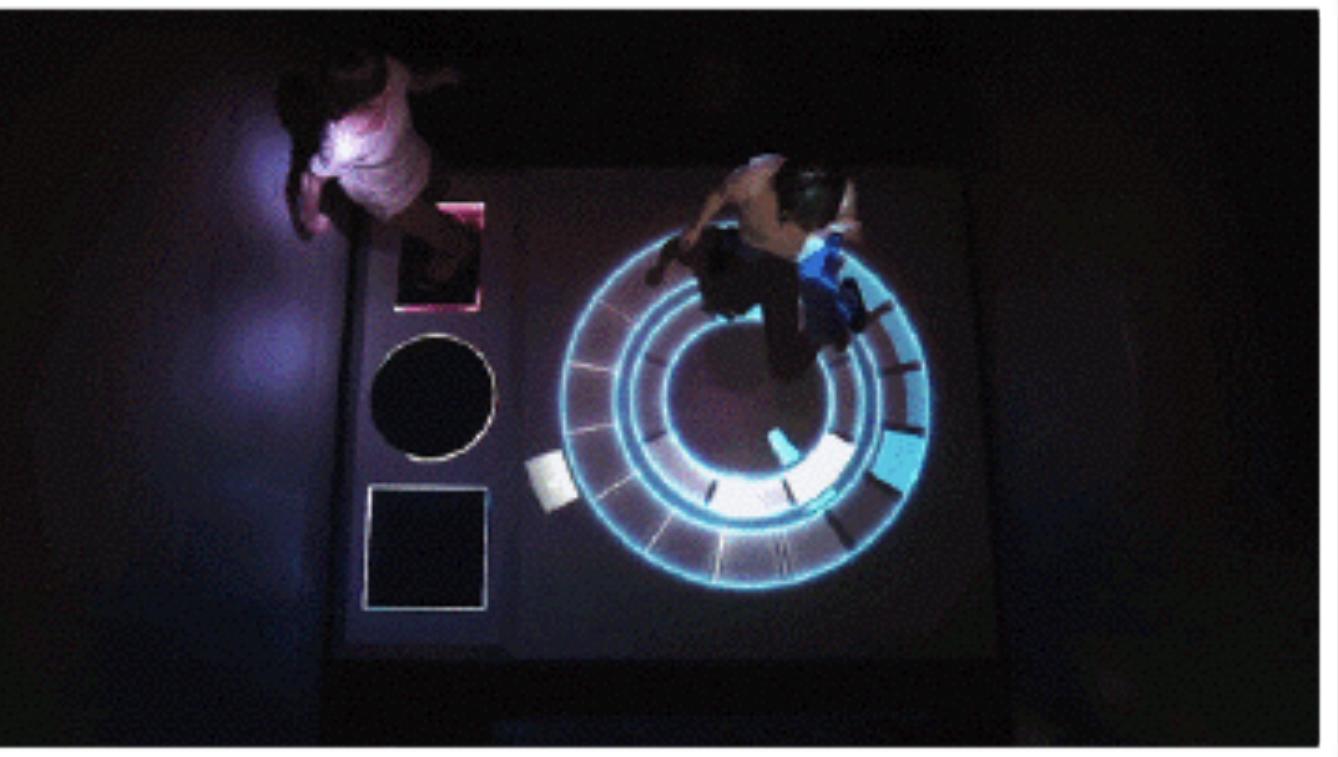
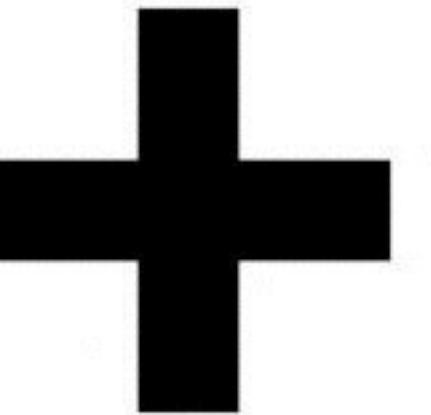
Responsive



Frontend réactif



Frontend réactif



Demo

Realtime web!





Et les perfs dans tout ca ?



SERLi



Et les perfs dans tout ca ?



SERLi



Les chiffres

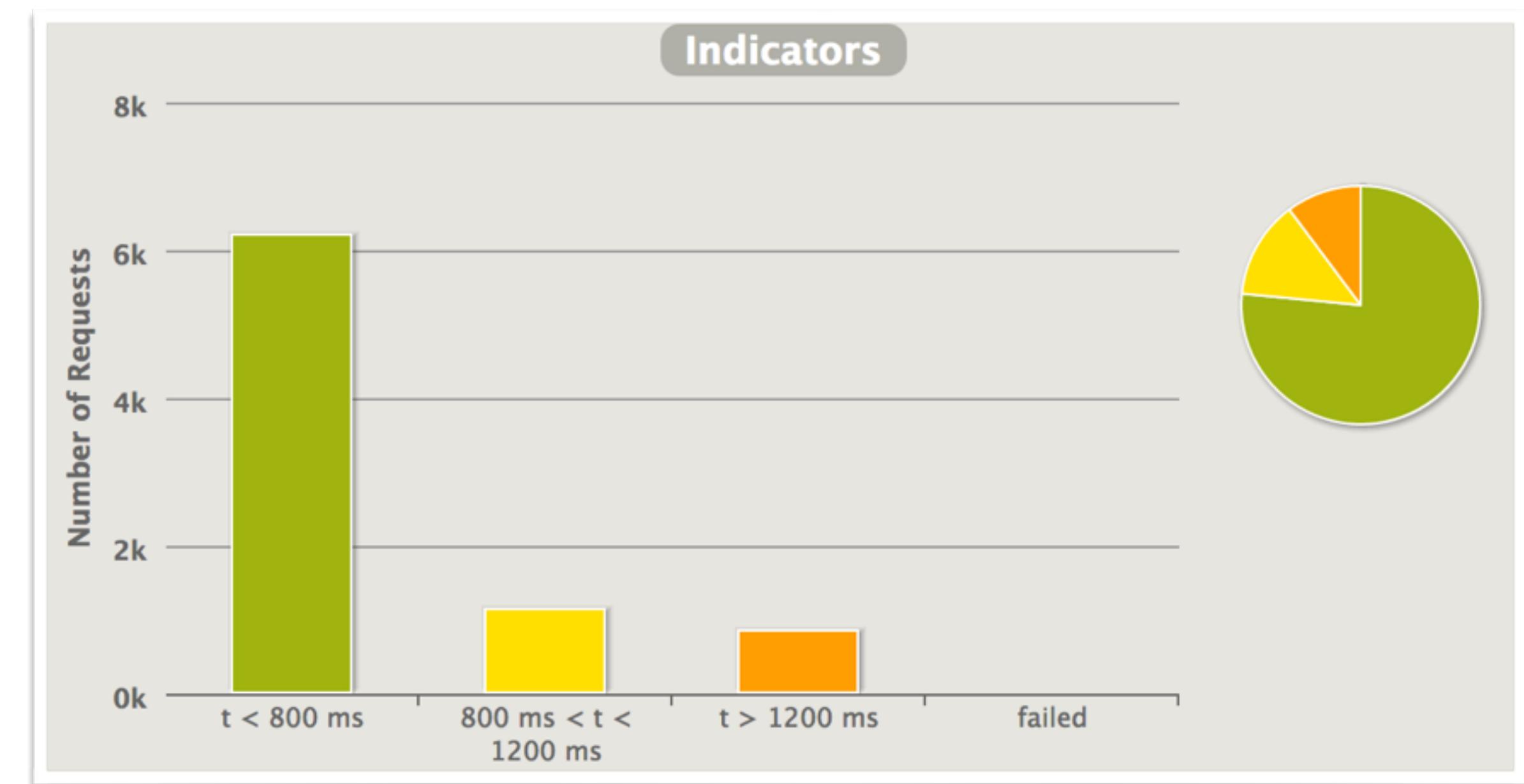
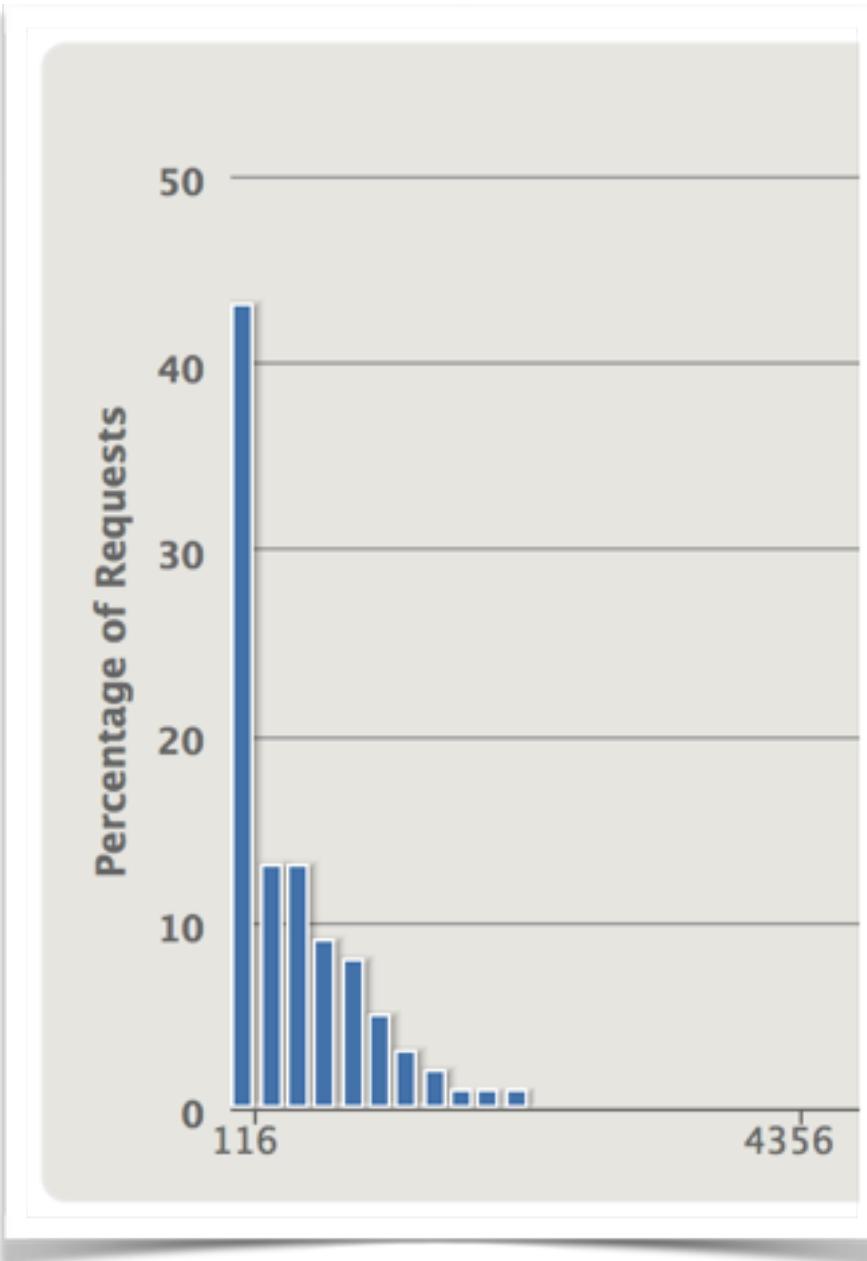
367 998 456 756



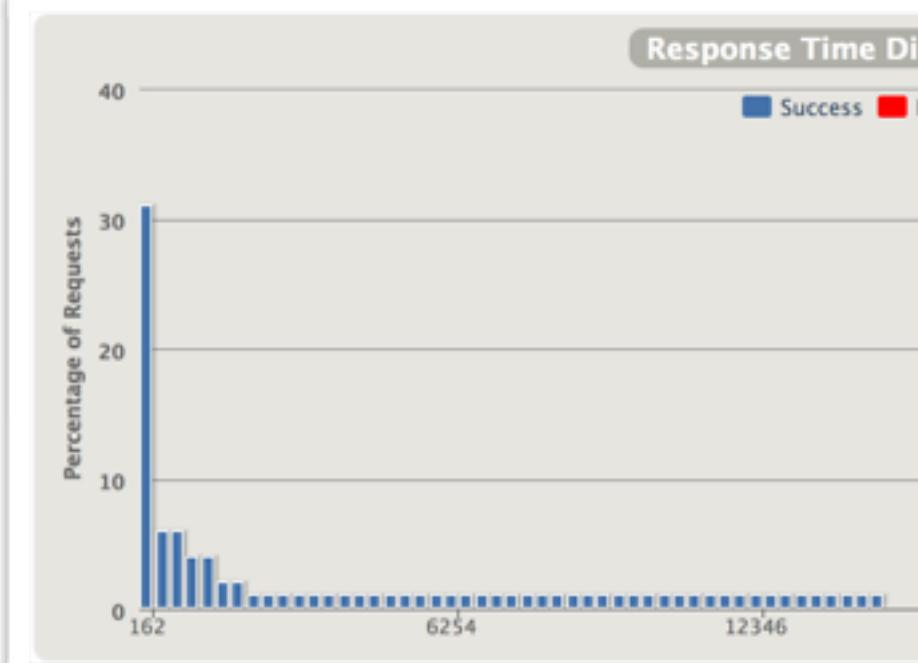
Les chiffres

367 998 456 756 €

200 clients



400 clients





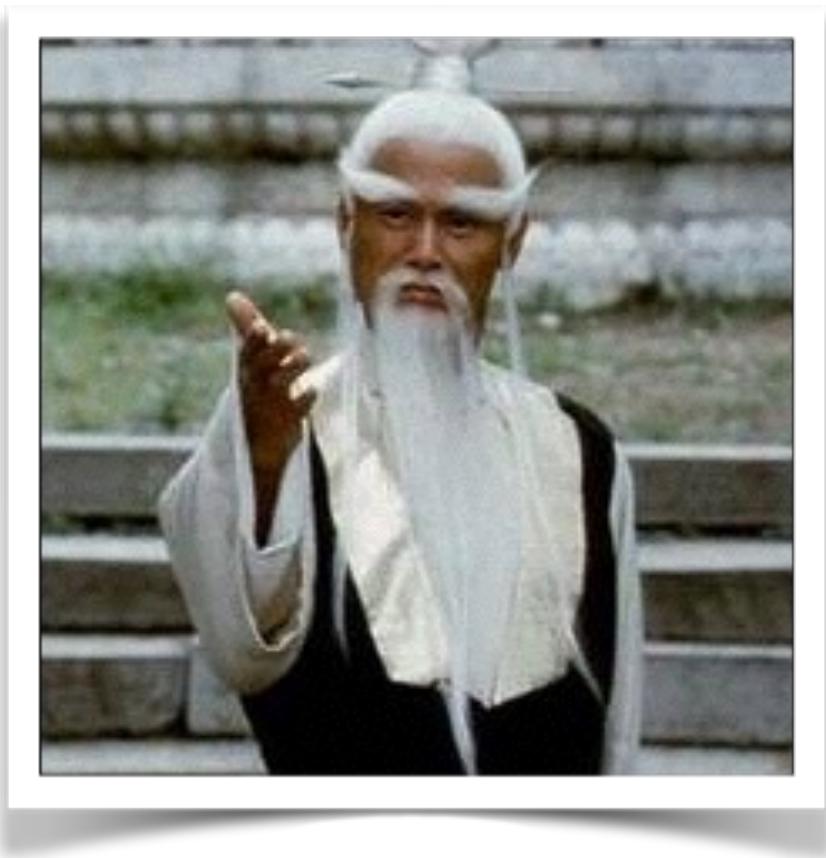
Metrics everywhere !!!



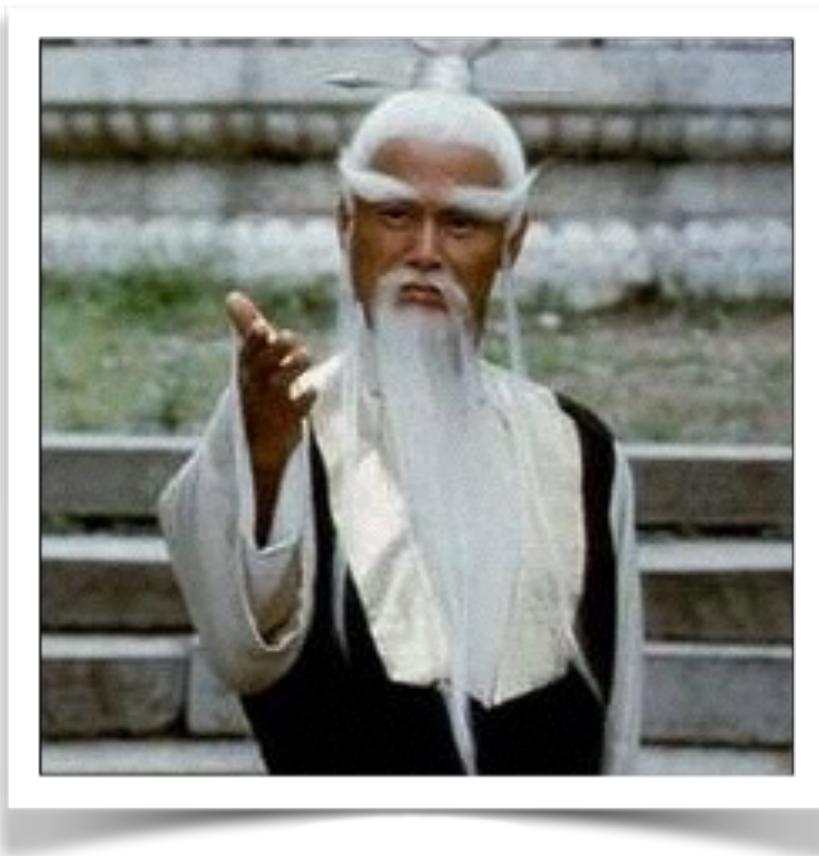
Les problèmes rencontrés



Nginx / mongo ...



Nginx / mongo ...



Cassandra



Cassandra



Bench web socket

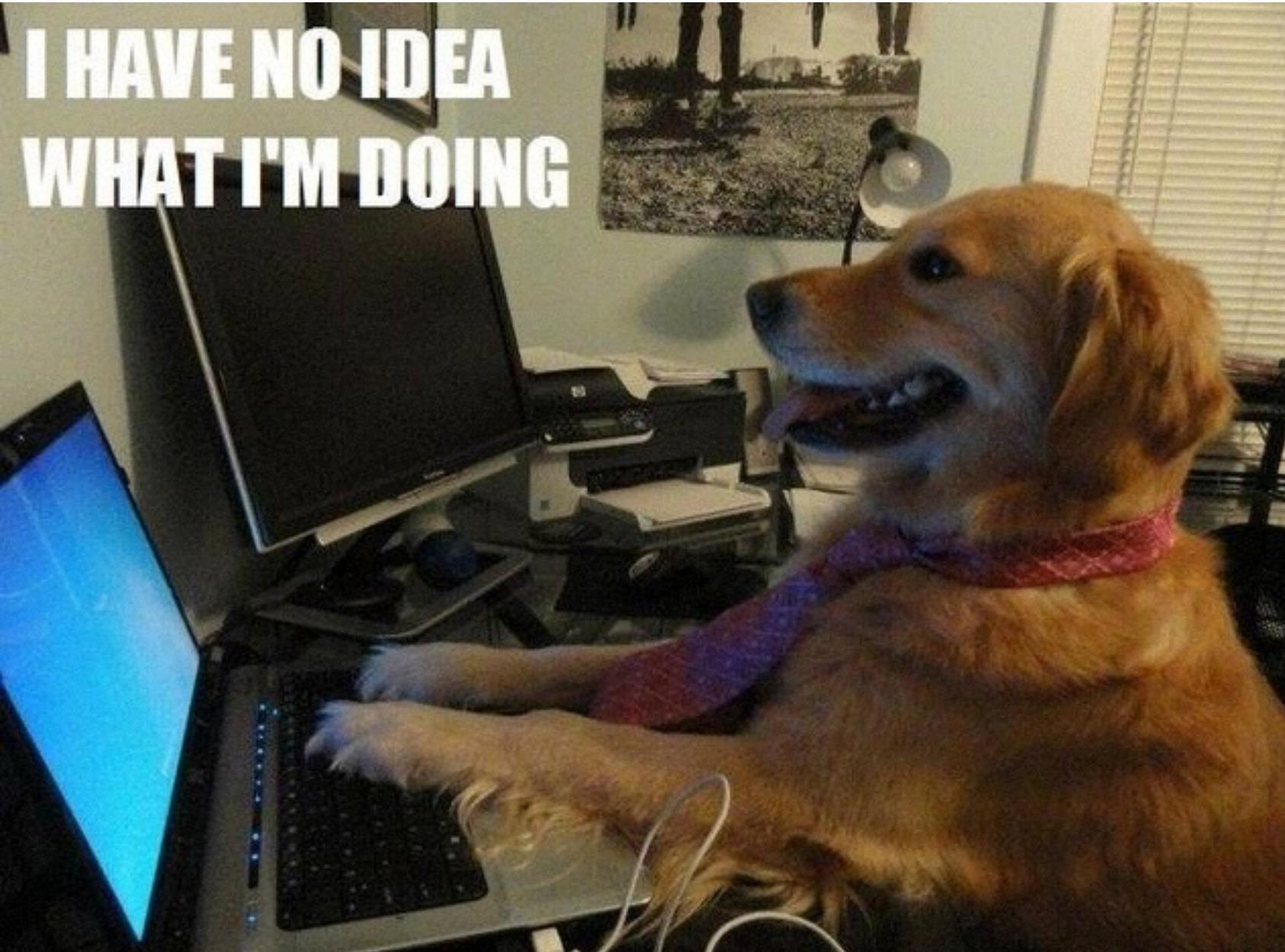


Bench web socket

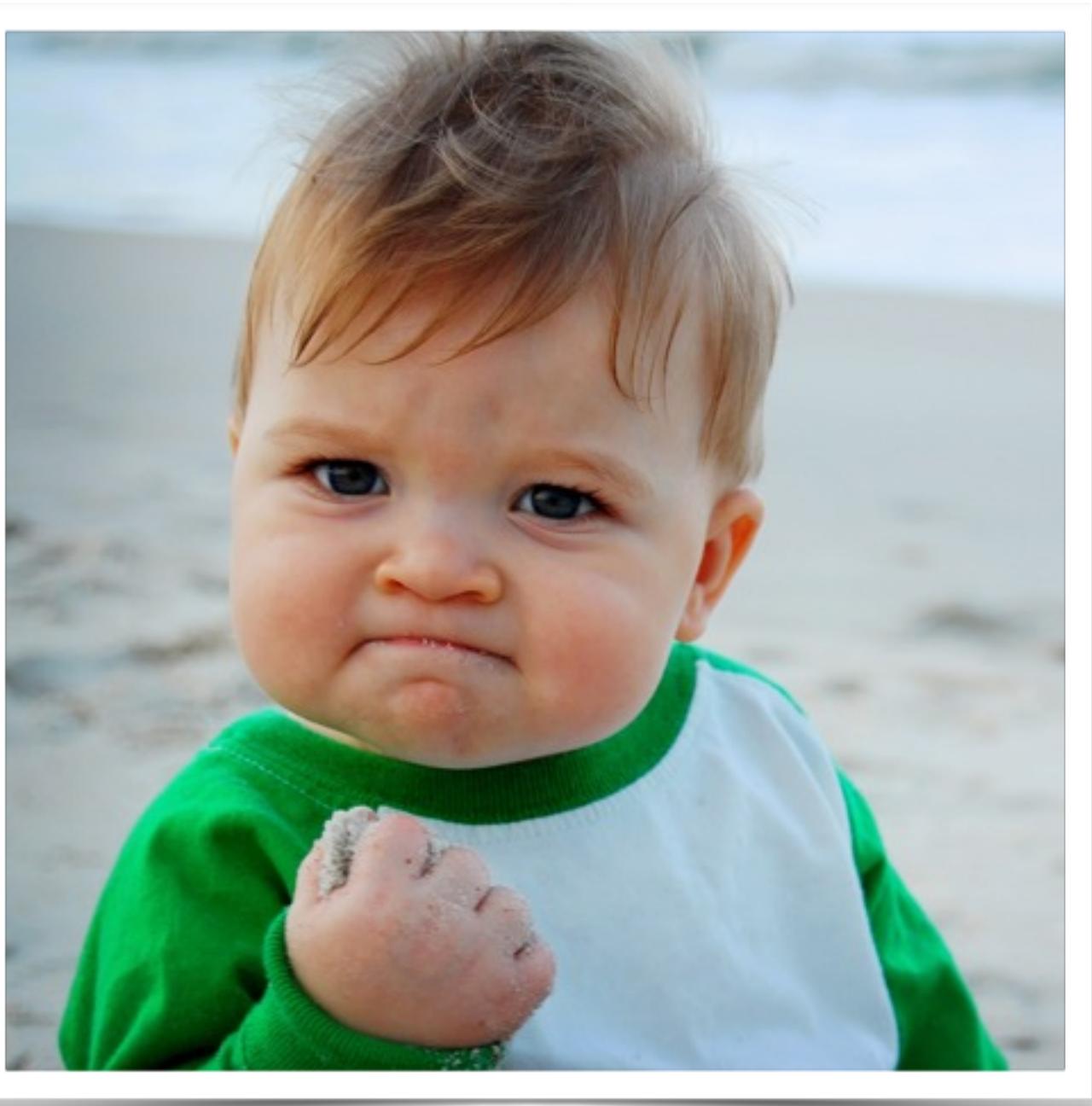




Configuration akka-cluster



Conclusion





scala



net



SERLi



scala



net



SERLi



This is the end ...