Highly available clustered apps on Amazon Web Services

Or how AWS saves us money

Tomasz Szymański @szimano
Adam Warski @adamwarski

softwaremill

Who are we?

- SoftwareMill
  - Extraordinary Software as as Standard
  - http://softwaremill.com
- Tomasz Szymański:
  - Warszawa-JUG leader
  - http://szimano.org
- Adam Warski:
  - Leader of Hibernate Envers, ElasticMQ
  - http://warski.org

Agenda

- Short intro: History of AWS, what is it?
- Demo1: app without the cloud
- Demo2: app with the cloud

softwaremill

Short history

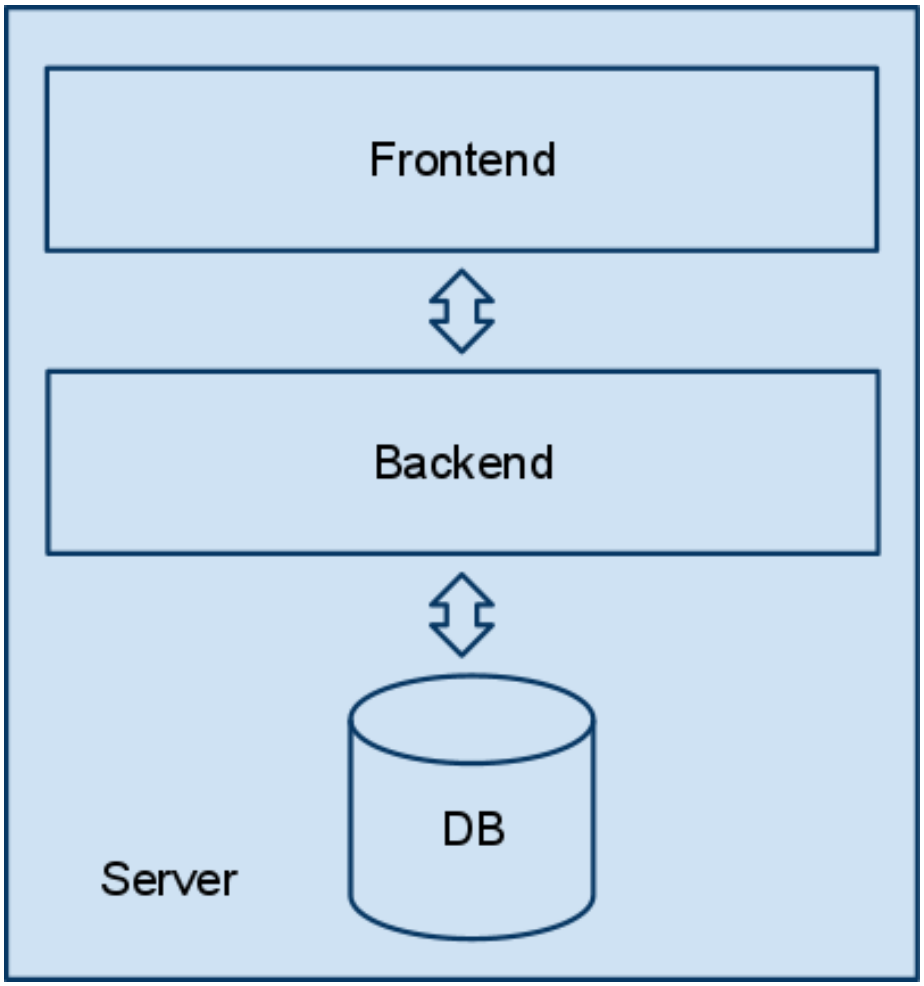
- .com boom – Amazon builds huge server centers in USA
- The bubble bursts – Amazon's need for the servers is much smaller than they originally though
- Amazon decides to sell computing power „per hour"
- July 2002 - Amazon Web Services starts

AWS – what is it?

- Set of different services:
  - EC2 – Elastic Compute Cloud
  - EBS – Elastic Block Store
  - S3 – Simple Storage Service
  - SQS – Simple Queue Service
  - SNS – Simple Notification Service
  - ELB – Elastic Load Balancing
  - DynamoDB/SimpleDB
  - RDS – Relational Database Service
- … and much much more

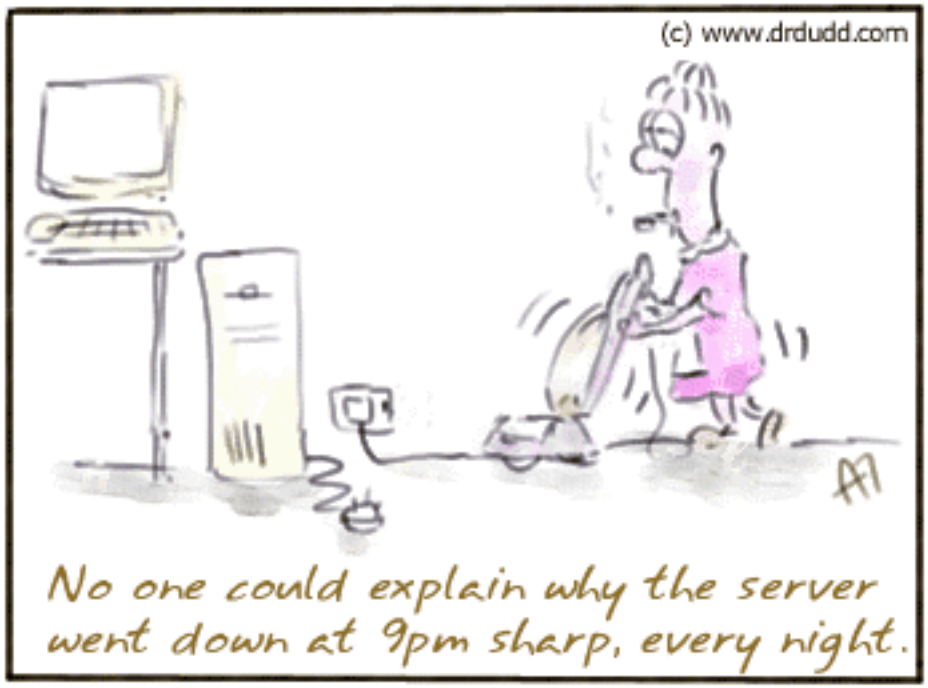
softwaremill

Our application



softwaremill

But...

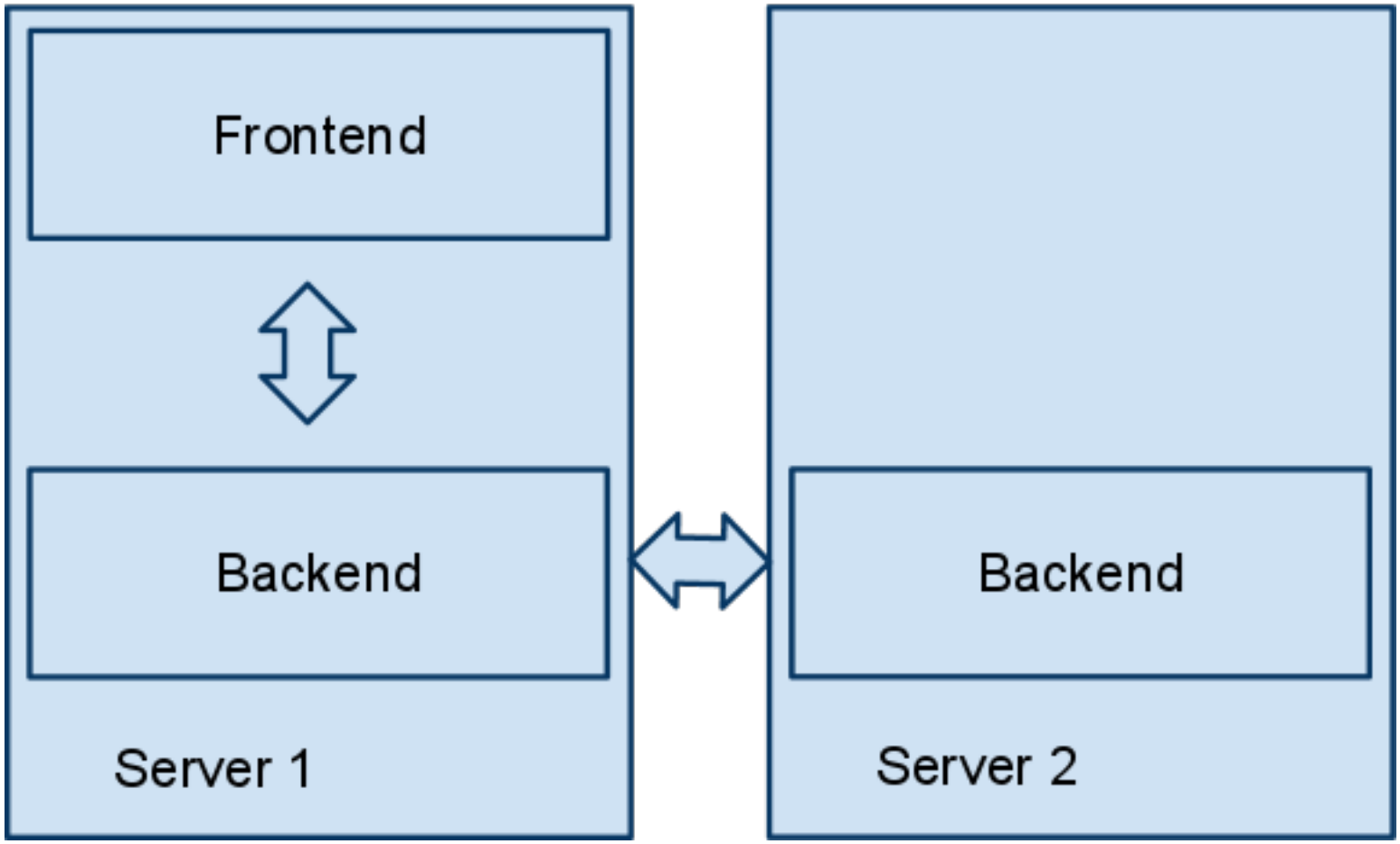


softwaremill

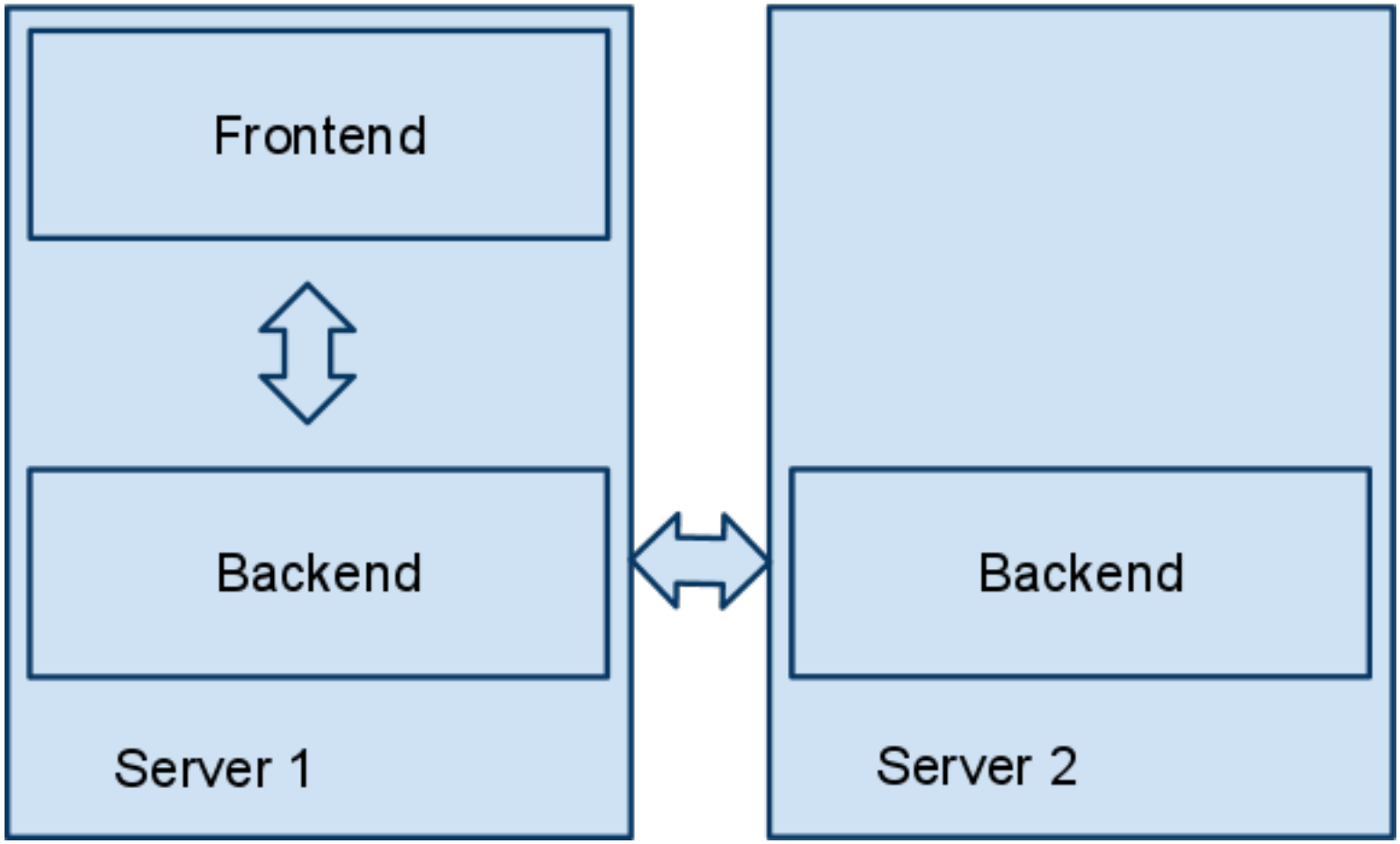
What do we want?

- Scalability
- High availability

Another server



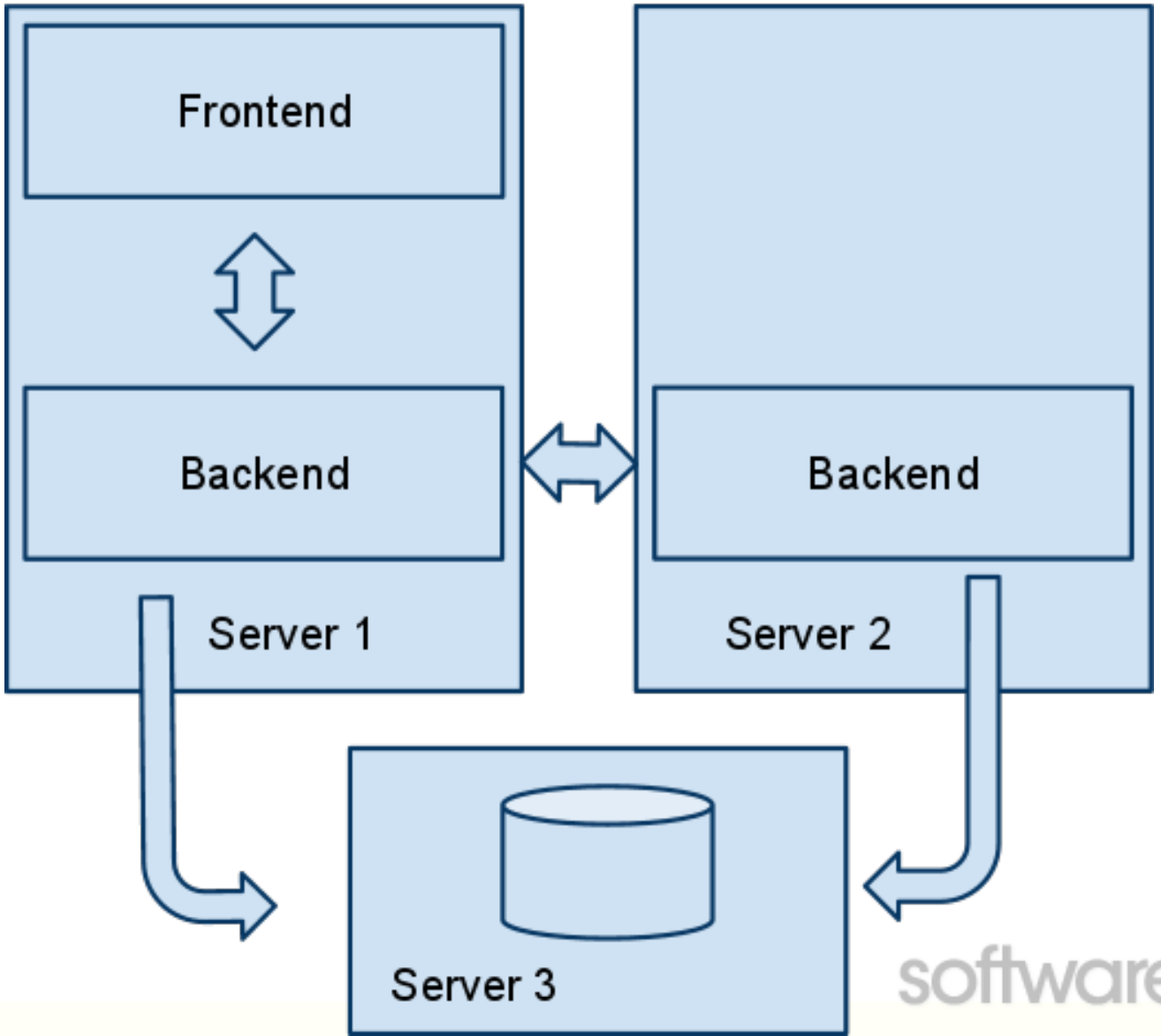
Another server



What about the DB?

softwaremill

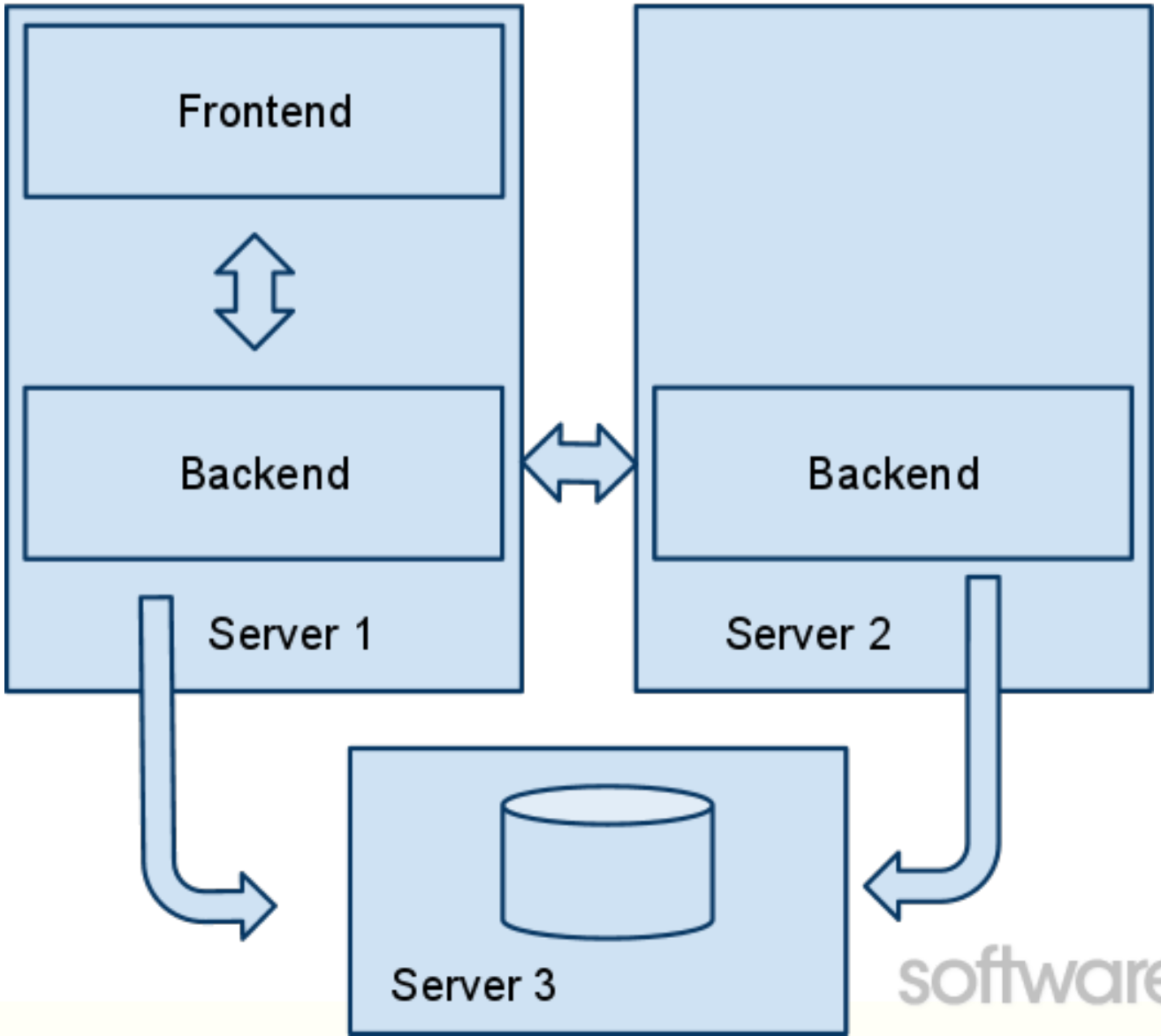
Data Base



softwaremill

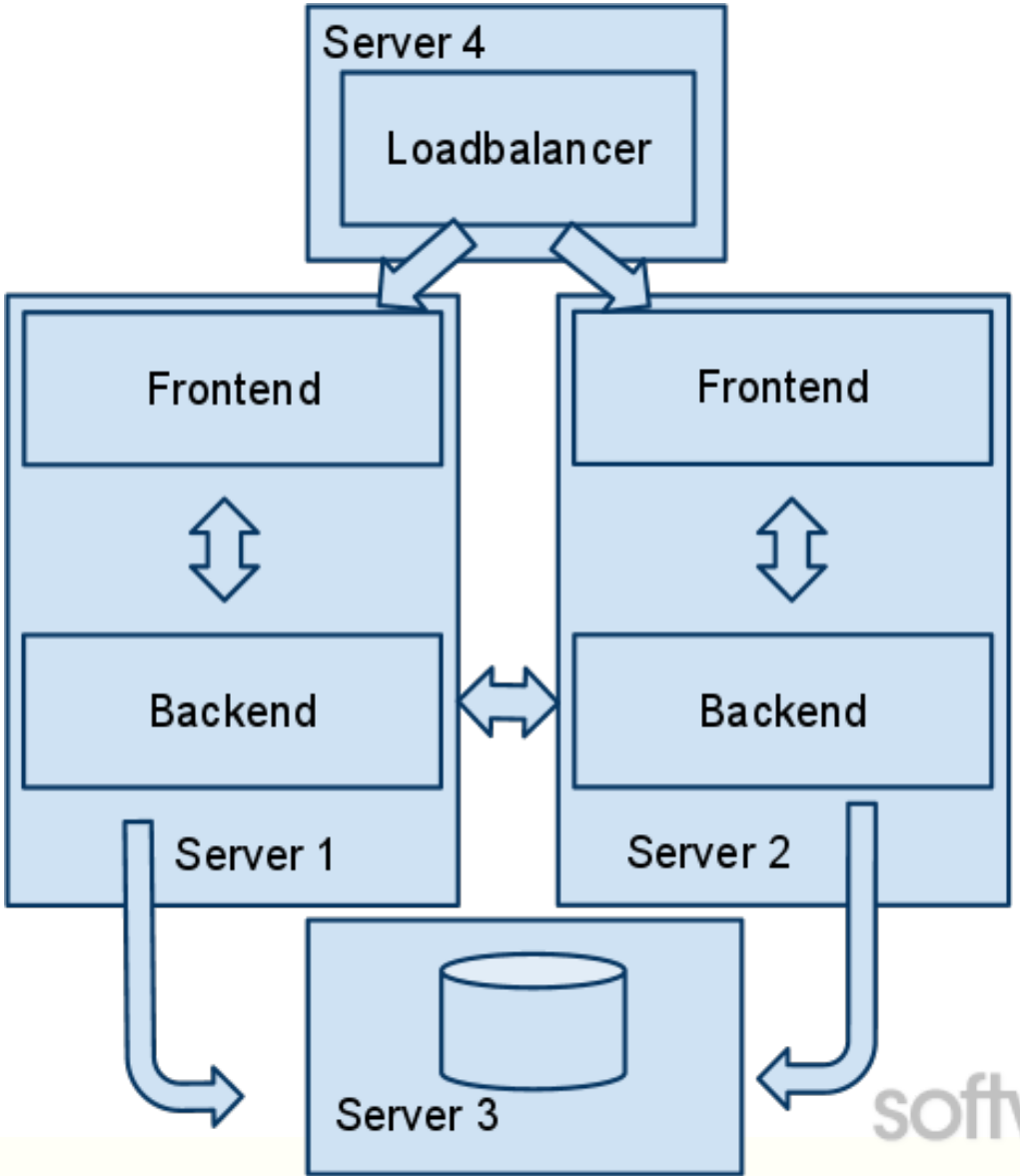
Data Base

What about scalability and HA of frontend?



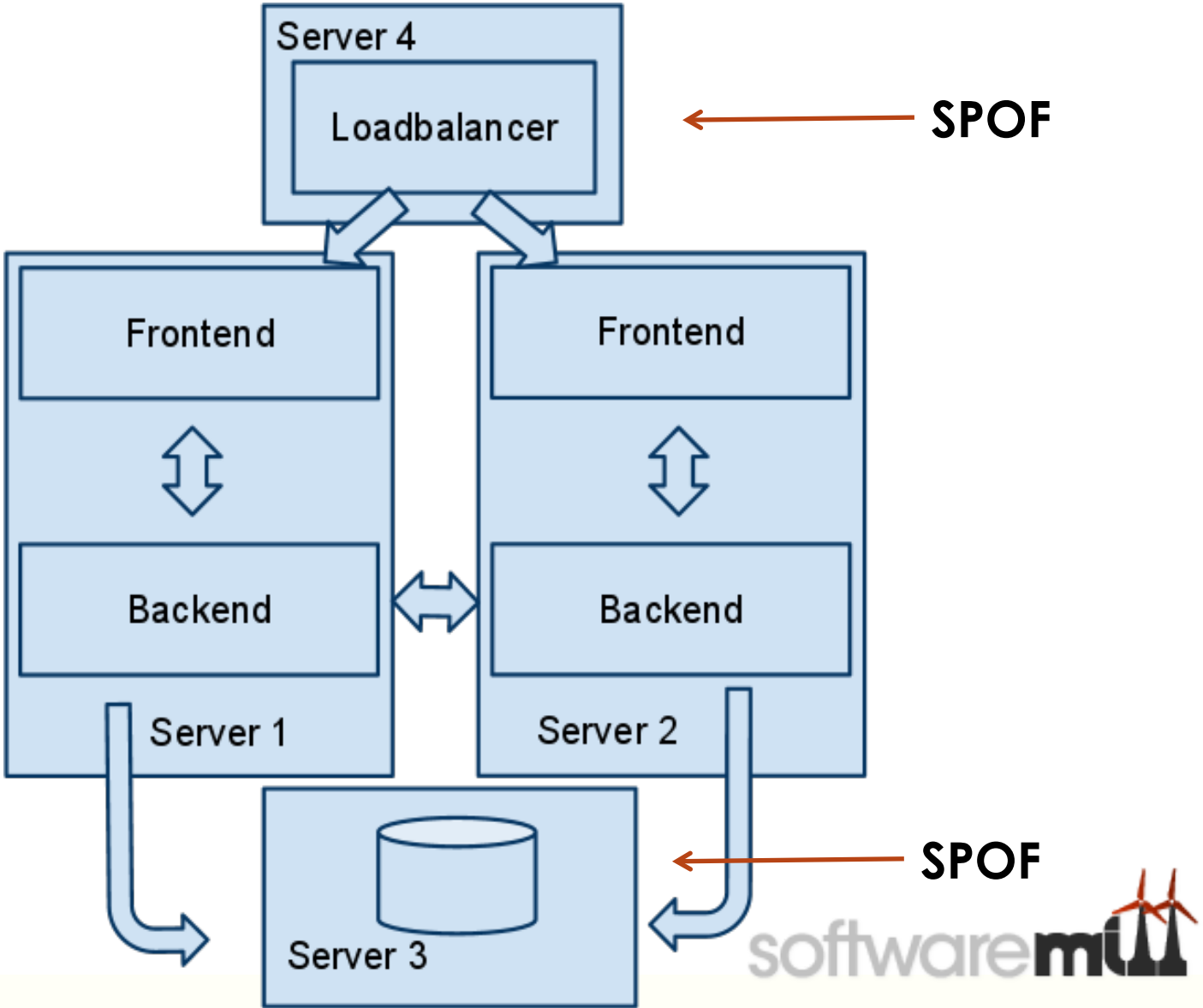
softwaremill

Frontend

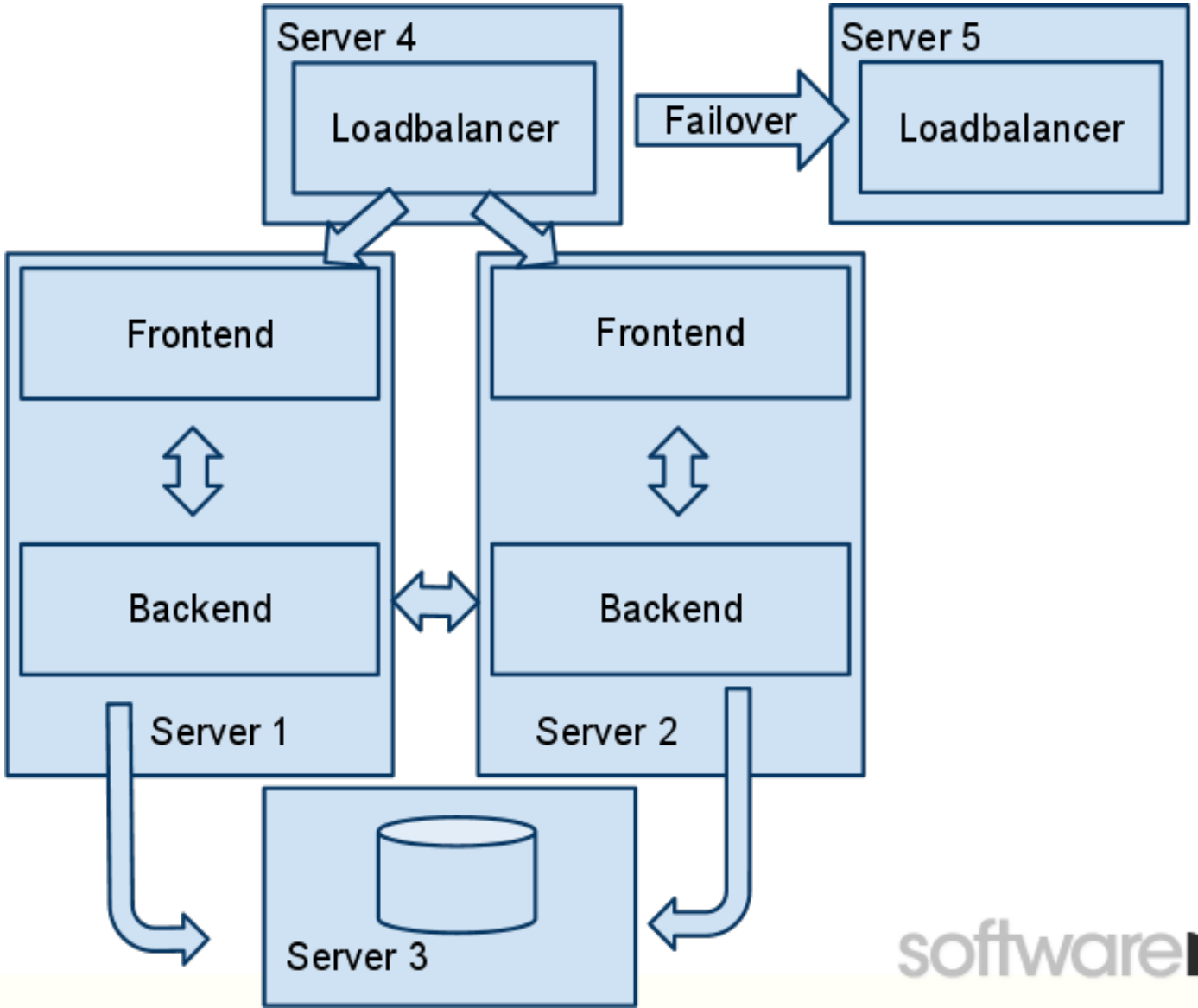


softwaremill

Frontend

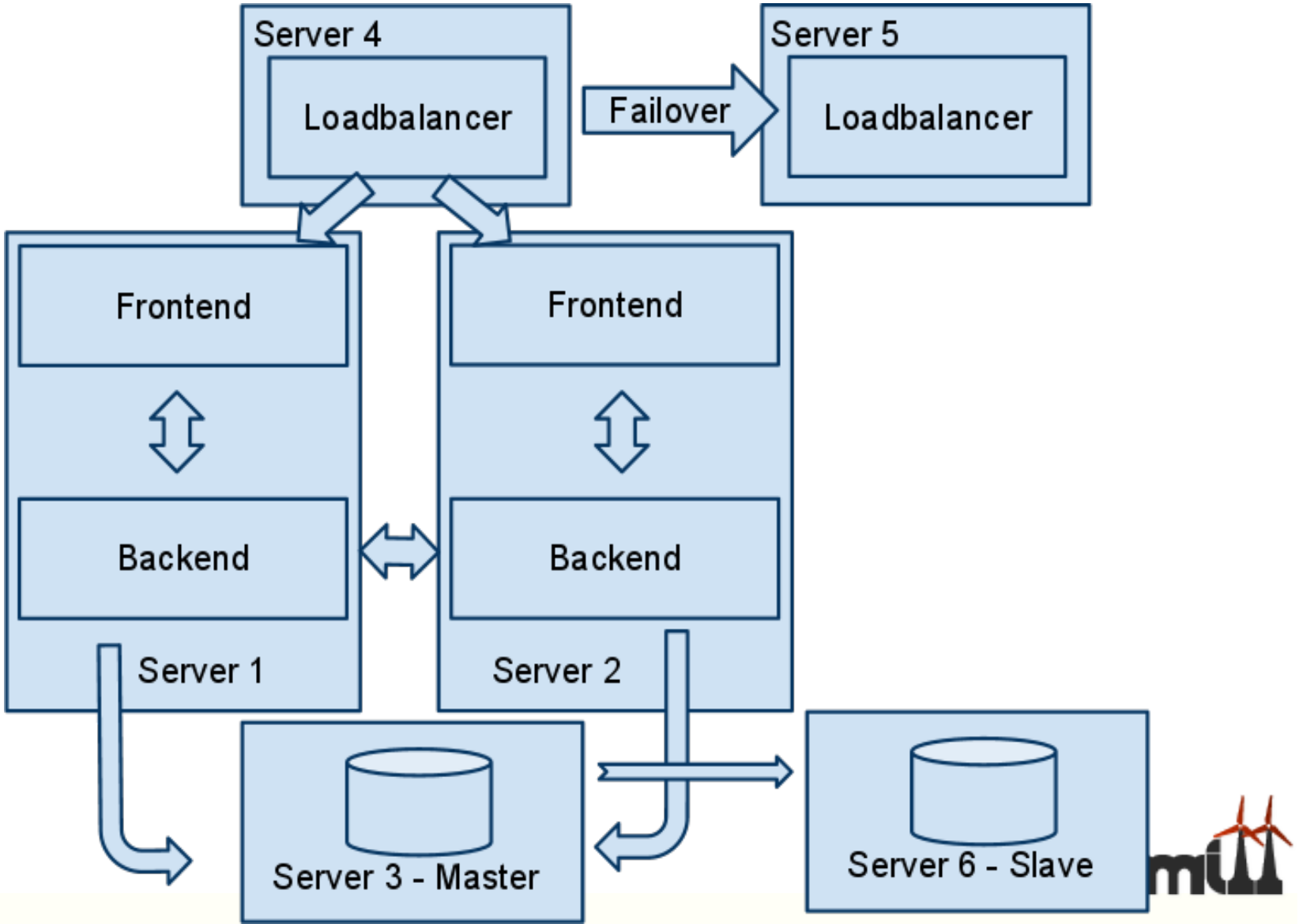

Server 4
Loadbalancer
SPOF
Frontend
Frontend
Backend
Backend
Server 1
Server 2
Server 3
SPOF

Loadbalancer



softwaremill

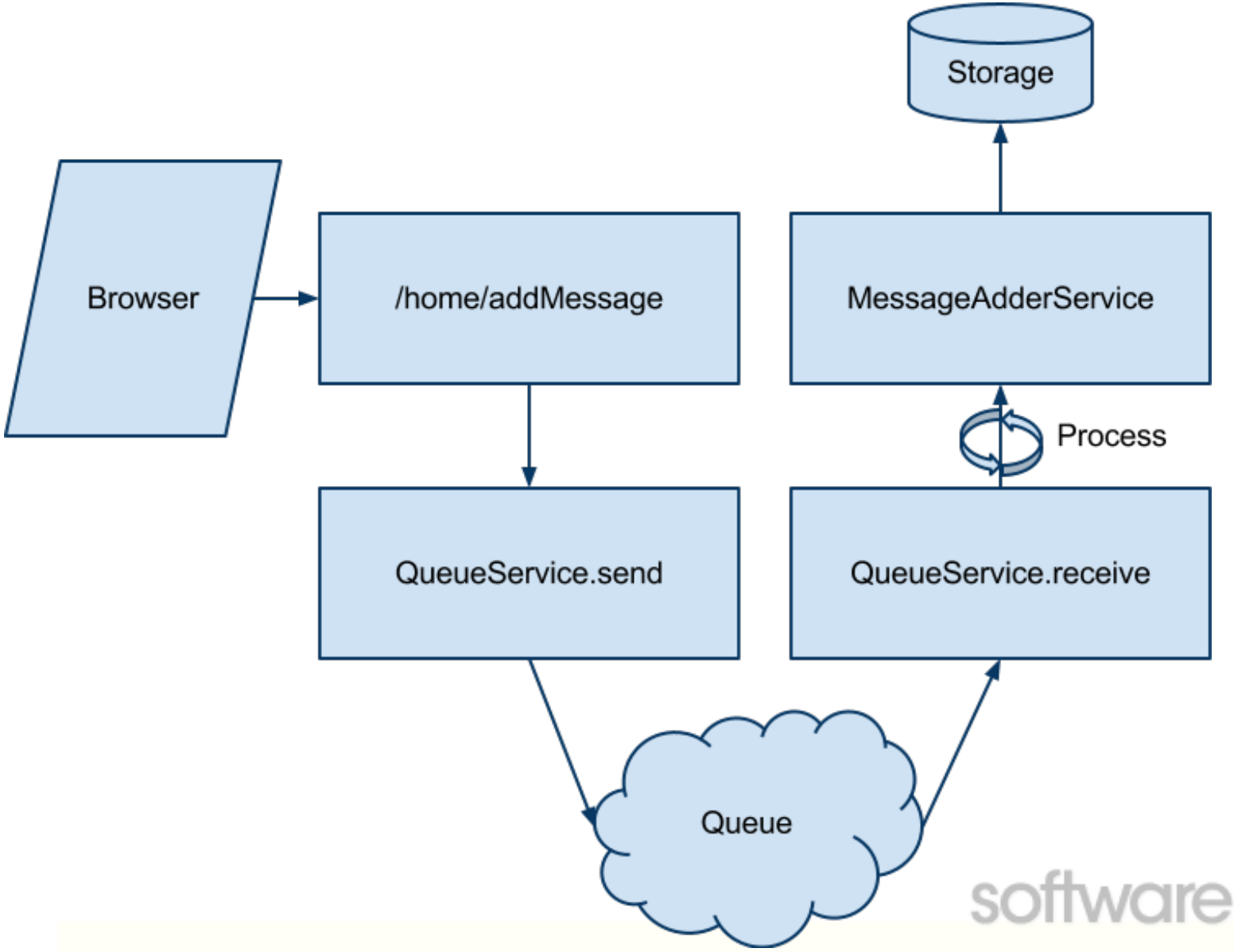
DB


Server 4
Loadbalancer
Failover
Server 5
Loadbalancer
Frontend
Frontend
Backend
Backend
Server 1
Server 2
Server 3 - Master
Server 6 - Slave

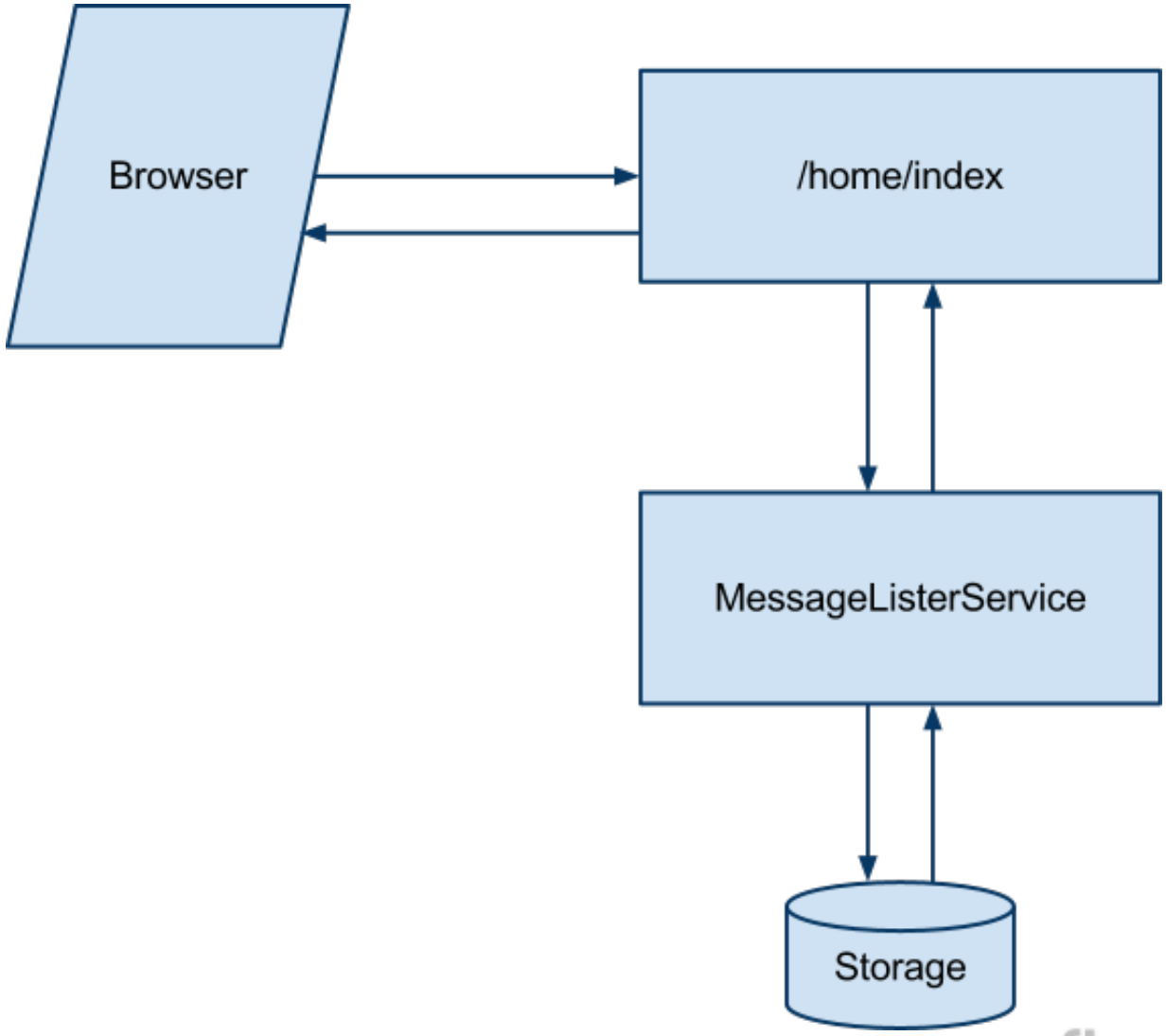
Effect?

- 6 servers
- Lots of money spent
- It still isn't very safe (we rely on one failover for each component)

Our application - write



Our application - read



softwaremill

Demo - locally

- JMS
- Hibernate
- H2

Servers – EC2

- New server in a of couple minutes
- API
- Lots of ready system images
- You can create your own images (AMI)
- Servers in different sizes and flavors
  - ie. small server 0.09c/h ~= 65 USD/month
- 7 regions, 2-4 availability zones in each
- SLA 99.95%

softwaremill

Servers – ELB

- You can attach as many EC2 servers as you want
- Works per region
  - Which means couple of availability zones
- Integration with Route53 (DNS) – domain support
- Sticky sessions
  - For example using JSESSIONID cookie

Persistence – SimpleDB

- NoSQL
- Key-value storage (multiple values per key)
- Eventual consistency, but:
  - Consistent read
  - Conditional put (~ transactions)
- SQL-like language for queries, but:
  - No JOINs
  - Query on N attributes will perform N queries and combine
- Everything is a String
- DynamoDB: successor

softwaremill

Queues - SQS

- Pull-type message receiving
- No transactions
- Message will be delivered again, if it's not deleted after receiving it
- Messages are stored on many servers, so there is a slight chance they can be delivered more than once
- ElasticMQ: implements SQS interface

softwaremill

Demo – Amazon

- SQS instead of JMS
- SDB instead of Hibernate
- EC2 instead of MacBook Pro
- All the code available on GitHub!
  - https://github.com/adamw/devcrowd-aws-demo

Links

- **https://github.com/adamw/devcrowd-aws-demo**
- **http://aws.amazon.com**
- https://www.jbison.com
- https://www.circulardms.com

softwaremill

Thank you

- Tomasz Szymański
  - tomasz.szymanski@softwaremill.com
  - @szimano
- Adam Warski
  - adam.warski@softwaremill.com
  - @adamwarski

Oh BTW - we are hiring :-)

softwaremill