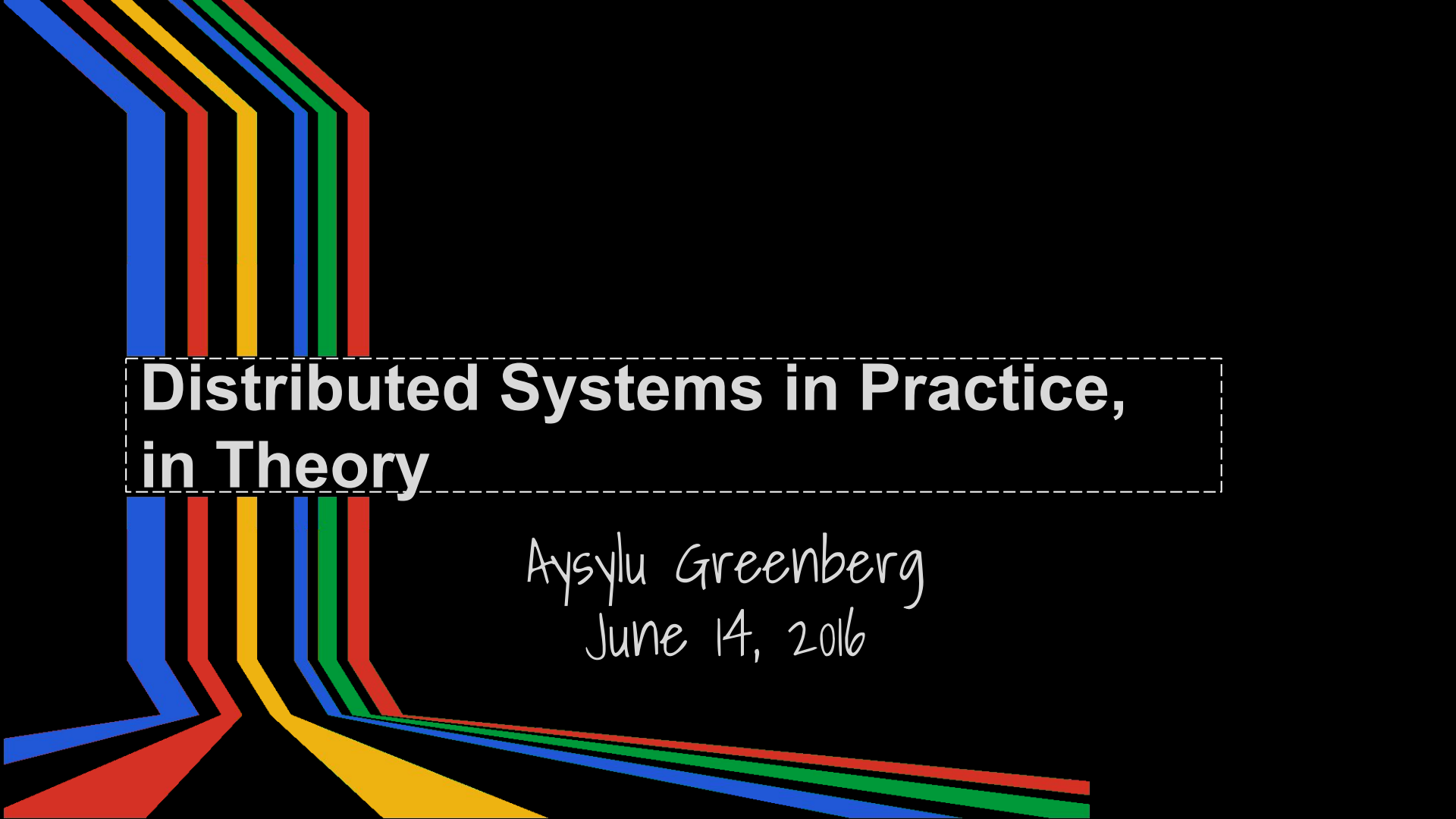

Distributed Systems in Practice, in Theory
Aysylu Greenberg
June 14, 2016

How I got into reading papers as a practitioner in industry

Computer Science
Research
In
Distributed Systems
Industry

Operating systems research

AN EXPERIMENTAL TIME-SHARING SYSTEM

Fernando J. Corbató, Marjorie Merwin Daggett, Robert C. Daley

Computation Center, Massachusetts Institute of Technology

Operating systems research

AN EXPERIMENTAL TIME-SHARING SYSTEM

Fernando J. Corbató, Marjorie Merwin Daggett, Robert C. Daley

Computation Center, Massachusetts Institute of Technology

Operating systems research

Concurrency

COOPERATING
SEQUENTIAL PROCESSES

EDSGER W. DIJKSTRA

(1965)

Operating systems research

Concurrency

COOPERATING
SEQUENTIAL PROCESSES

EDSGER W. DIJKSTRA

(1965)

Concurrency primitives: mutex & semaphore

Operating systems research

Concurrency

COOPERATING
SEQUENTIAL PROCESSES

Processes execute at different speeds

Concurrency primitives: mutex & semaphore

Time in distributed systems

Time, Clocks, and the Ordering of Events in a Distributed System

Leslie Lamport
Massachusetts Computer Associates, Inc.

https://www.flickr.com/photos/national_archives_of_norway/6263353228

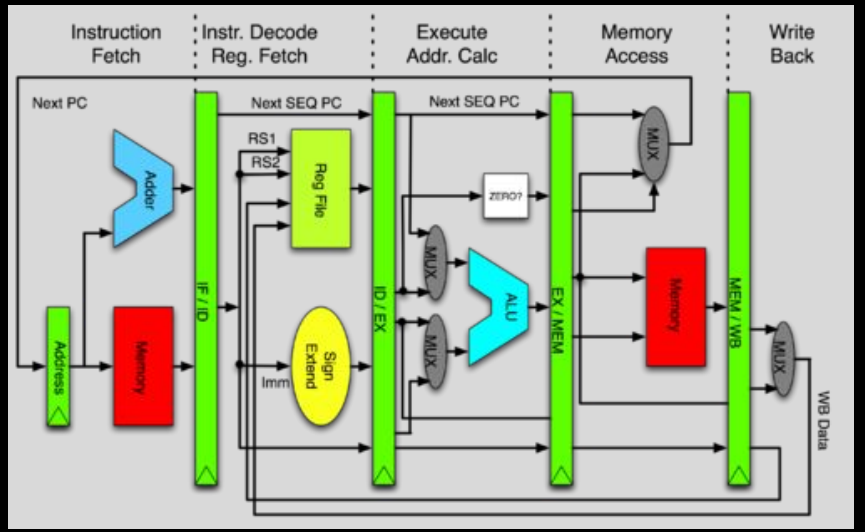
Time in distributed systems

Time, Clocks, and the Ordering of Events in a Distributed System

Leslie Lamport
Massachusetts Computer Associates, Inc.

Time in distributed systems

Pipelining



1980

1980

Internet

1980

Internet

Distributed consensus

1980

Internet

Distributed consensus

Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems

Brian M. Oki
Barbara H. Liskov

Massachusetts Institute of Technology

1980

Internet

Distributed consensus

Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems

Brian M. Oki
Barbara H. Liskov

Massachusetts Institute of Technology

The Part-Time Parliament

Leslie Lamport

1980

1980

Internet

Distributed consensus

Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems

Brian M. Oki
Barbara H. Liskov

Massachusetts Institute of Technology

The Part-Time Parliament

Paxos

Reconsider large systems

Reconsider large systems

Shared infrastructure

...

CS Research is Timeless

Inform decisions

Mitigate technical risk

Aysylu Greenberg

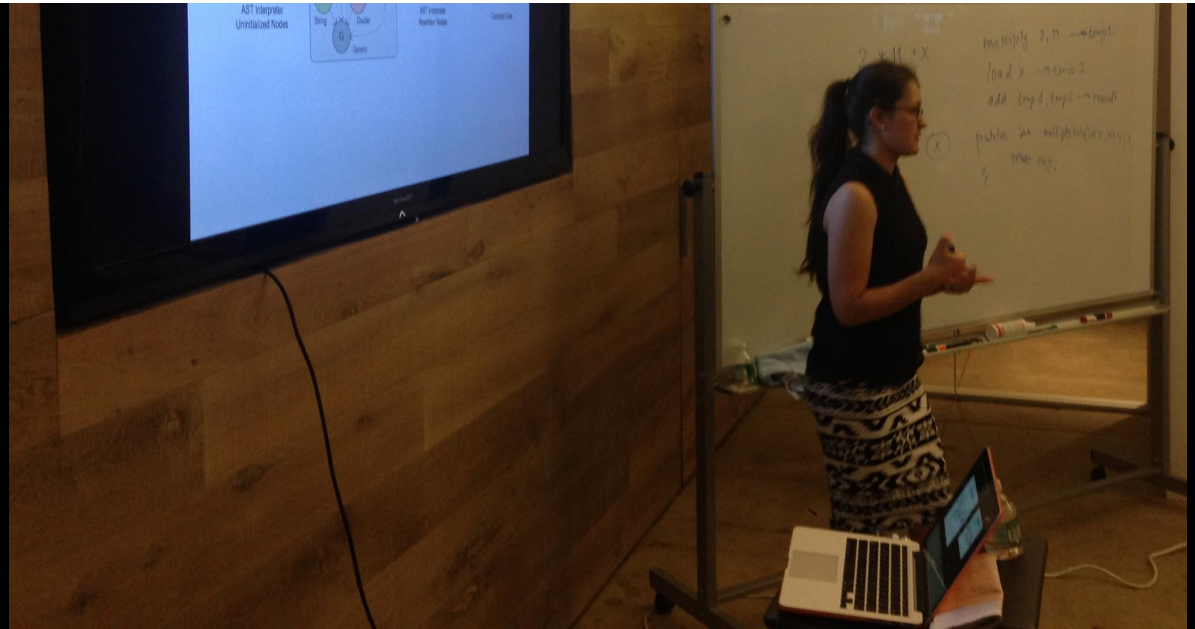
Google

@aysylu22

Papers We Love NYC

One VM to Rule Them All

Thomas Würthinger* Christian Wimmer* Andreas Wöß† Lukas Stadler†
Gilles Duboscq† Christian Humer† Gregor Richards§ Doug Simon* Mario Wolczko*



Papers We Love SF

Probabilistic Accuracy Bounds for Fault-Tolerant Computations that Discard Tasks *

Martin Rinard
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

Aysylu Greenberg

Google

$f(x) = x$

@aysylu22

Today

- Staged Event-Driven Architecture

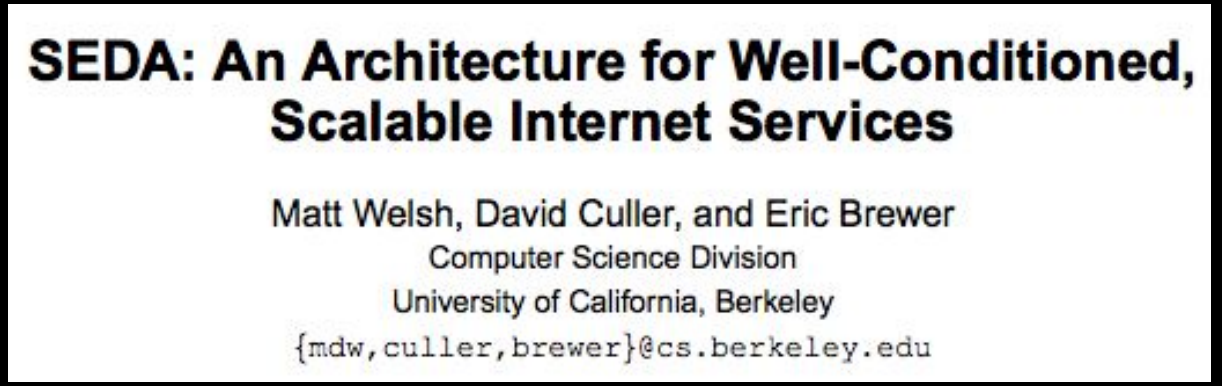
Today

- Staged Event-Driven Architecture
- Leases

Today

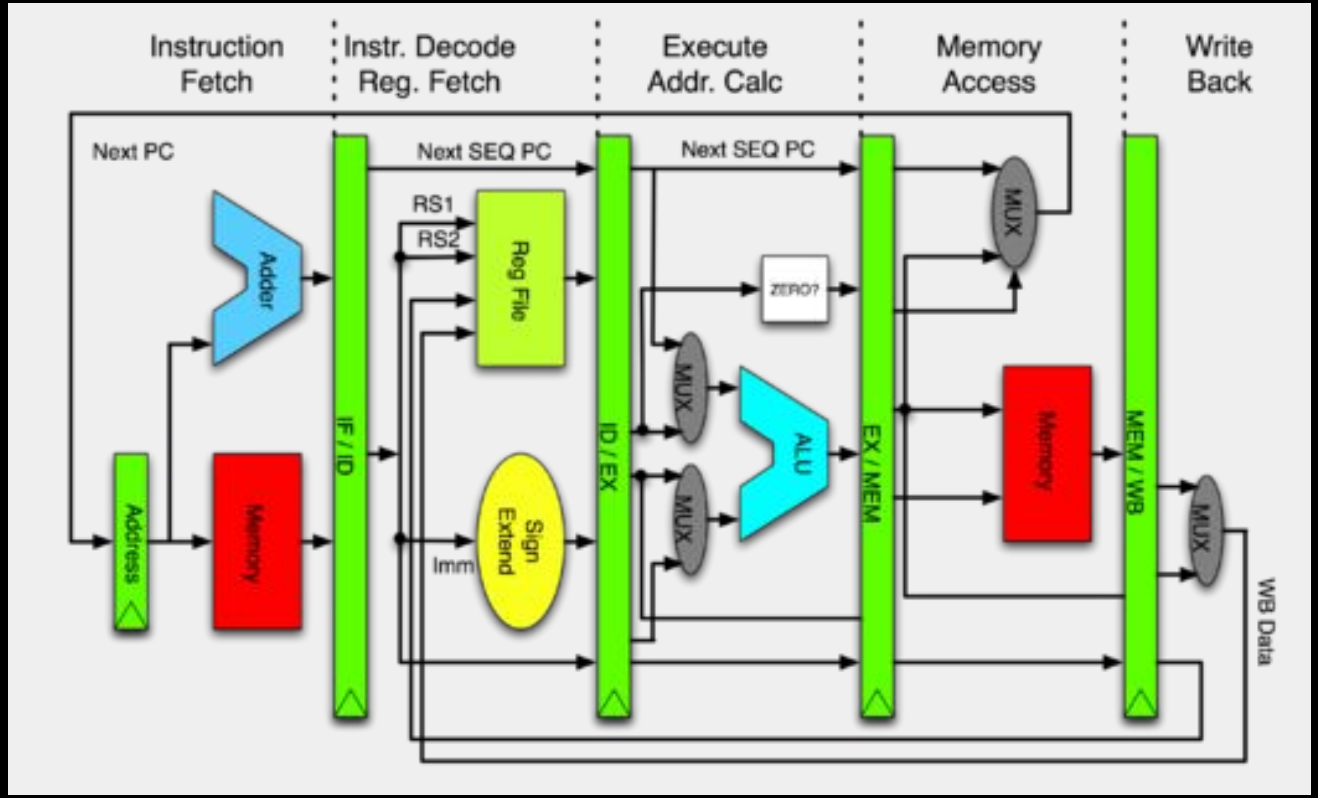
- Staged Event-Driven Architecture
- Leases
- Inaccurate Computations

Staged Event Driven Architecture & *Deep Pipelines*

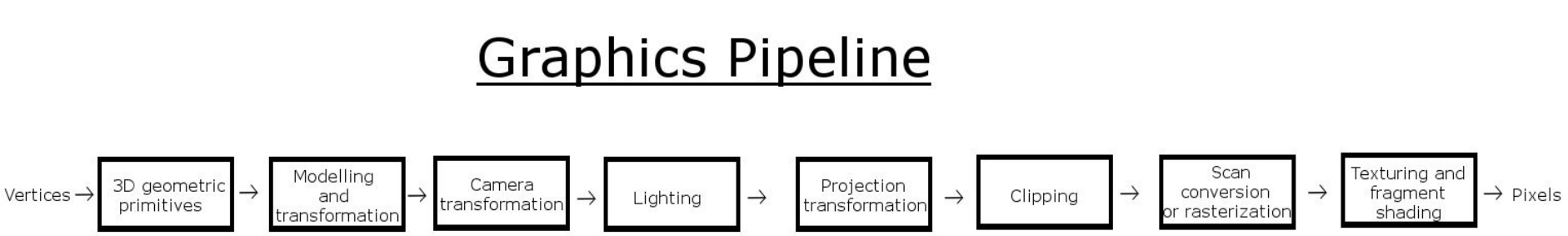


2001

Hardware to Data Pipelines



Hardware to Data Pipelines

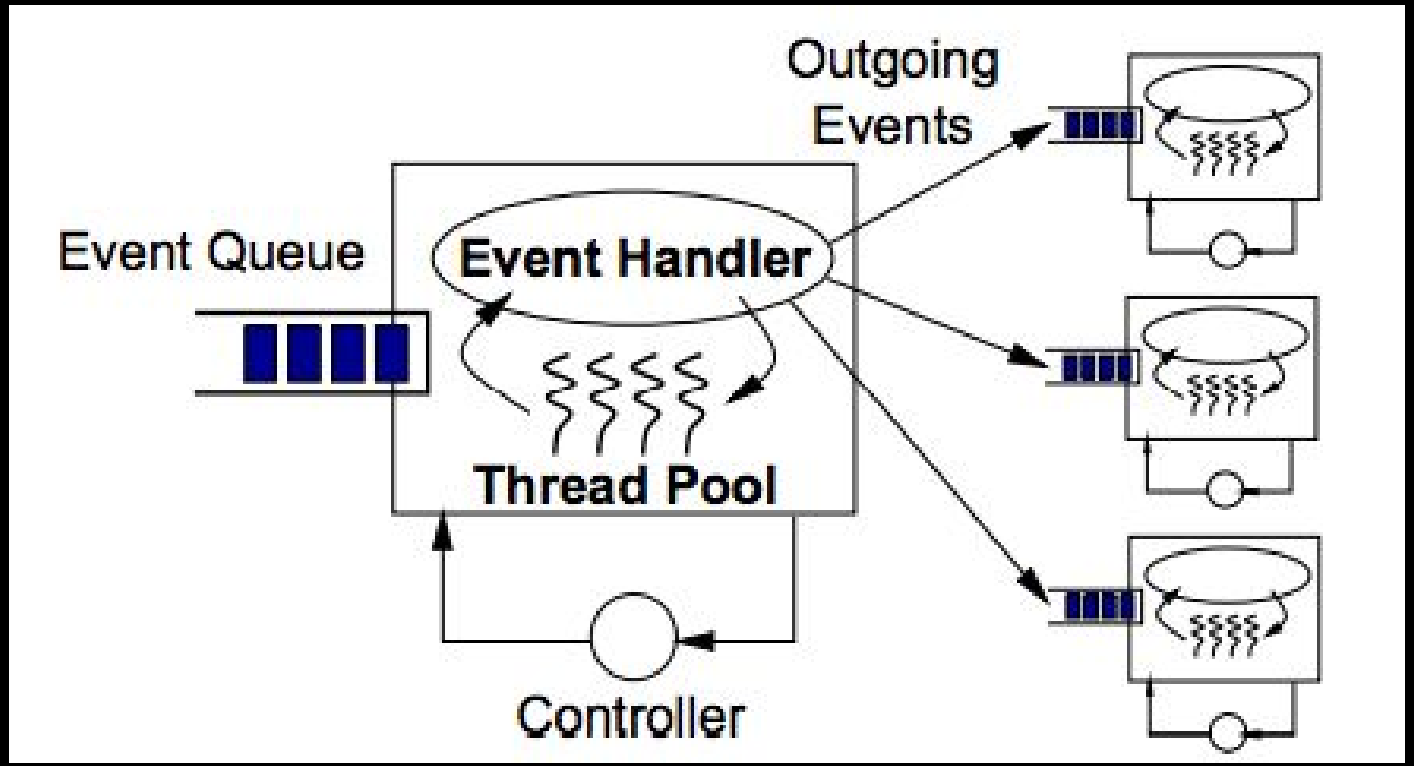


https://en.wikipedia.org/wiki/Graphics_pipeline

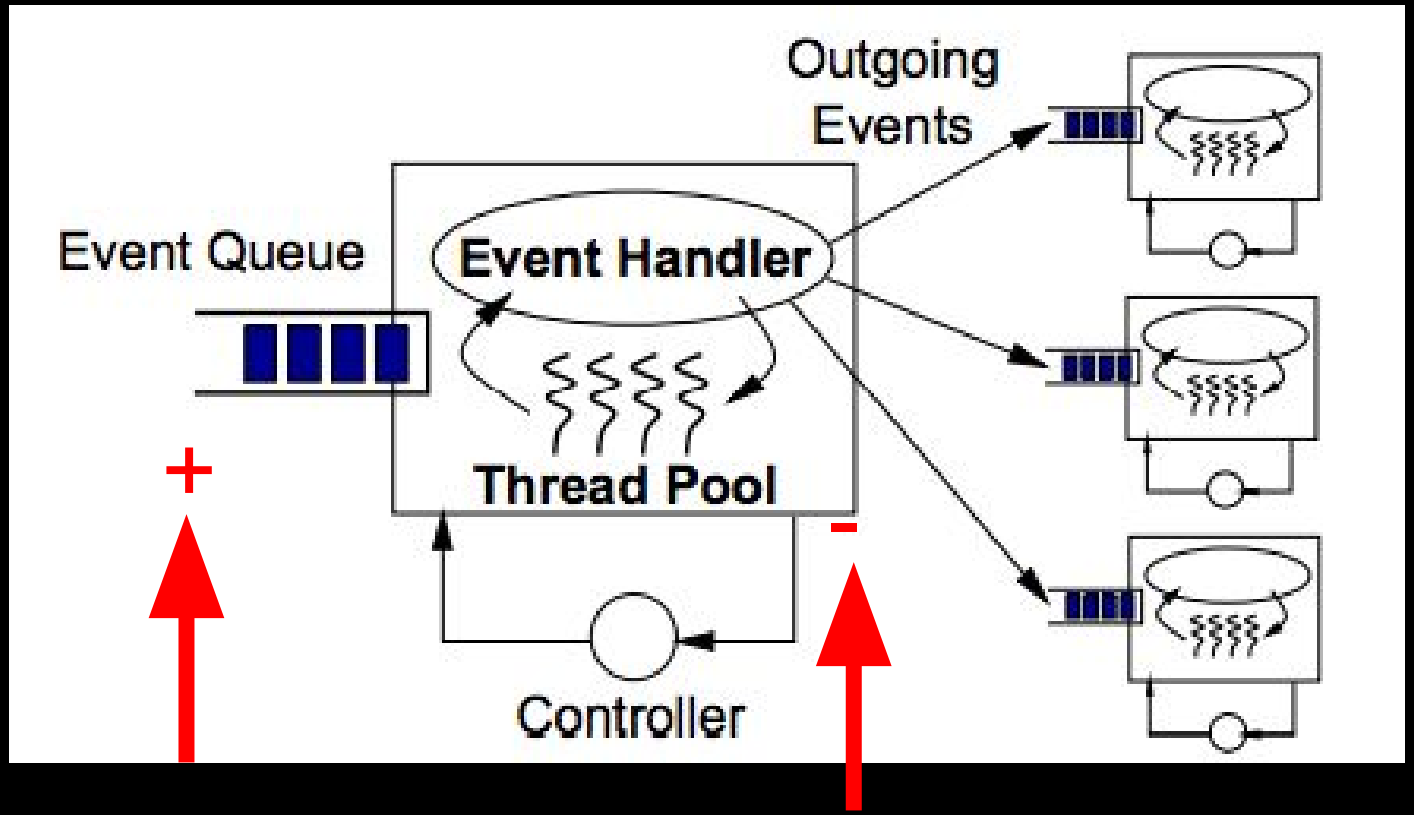
SEDA: An Architecture for Well-Conditioned, Scalable Internet Services

Matt Welsh, David Culler, and Eric Brewer
Computer Science Division
University of California, Berkeley
{mdw,culler,brewer}@cs.berkeley.edu

Staged Event Driven Architecture



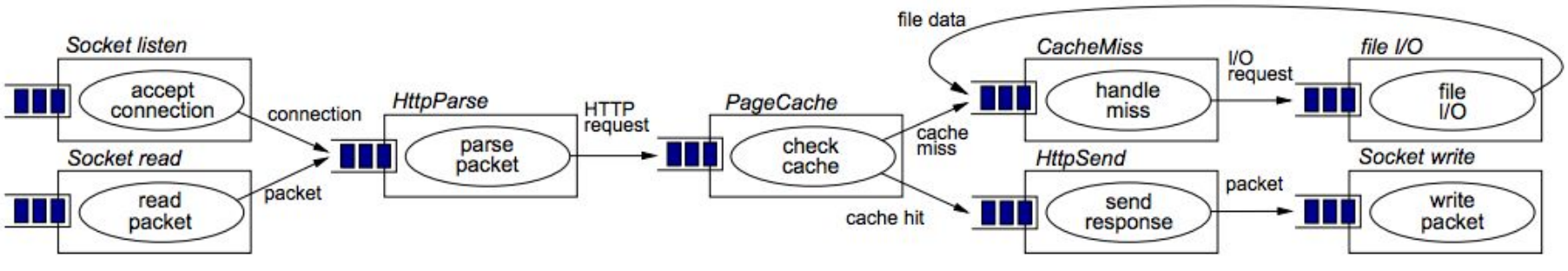
Staged Event Driven Architecture



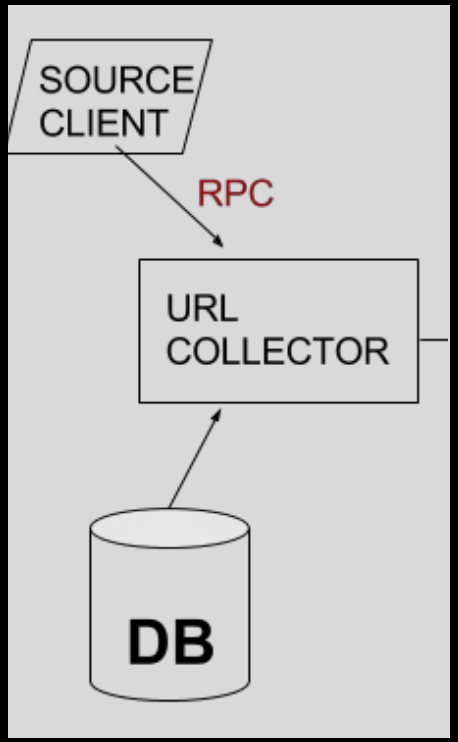
Staged Event Driven Architecture

Single-machine pipeline

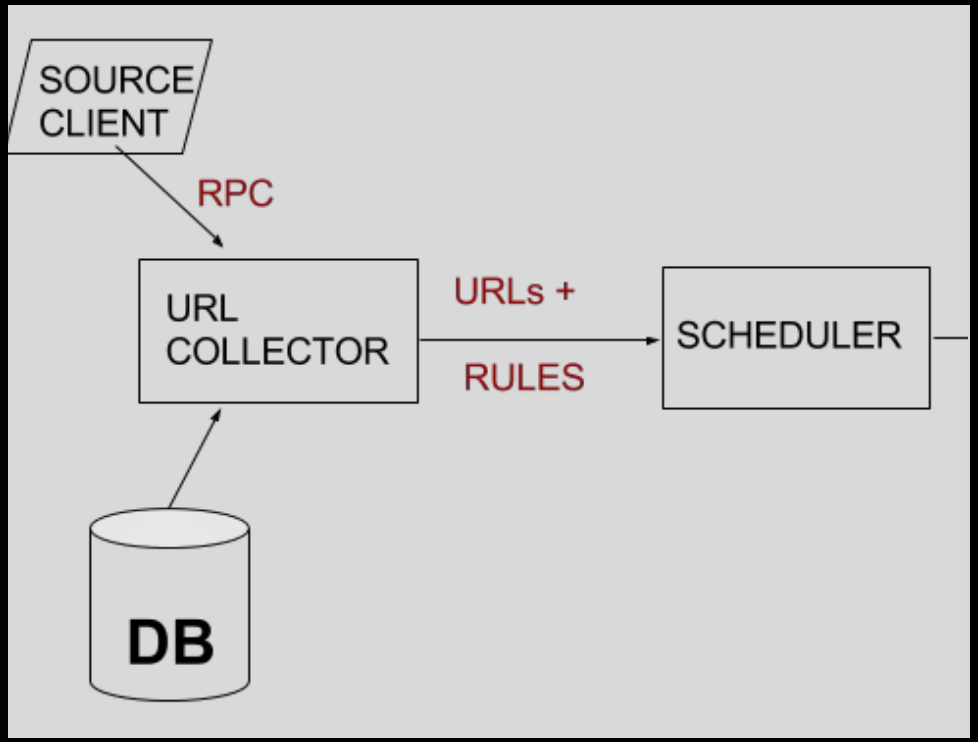
generalizes to distributed pipelines



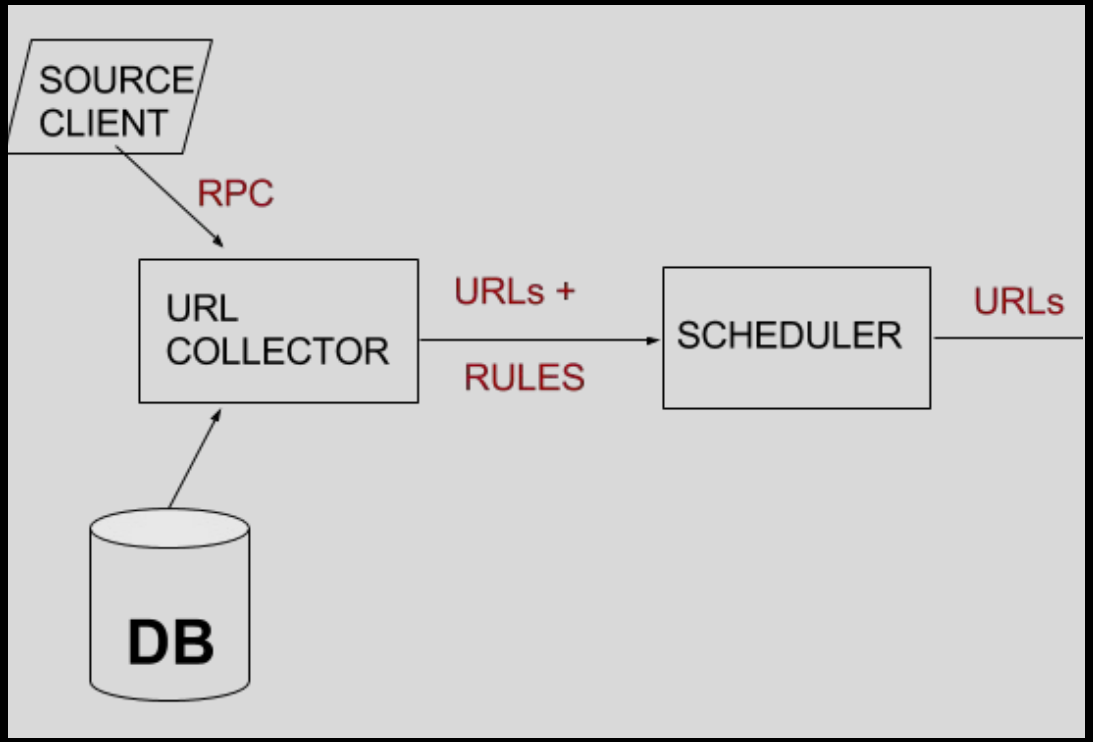
Search Indexing Pipelines



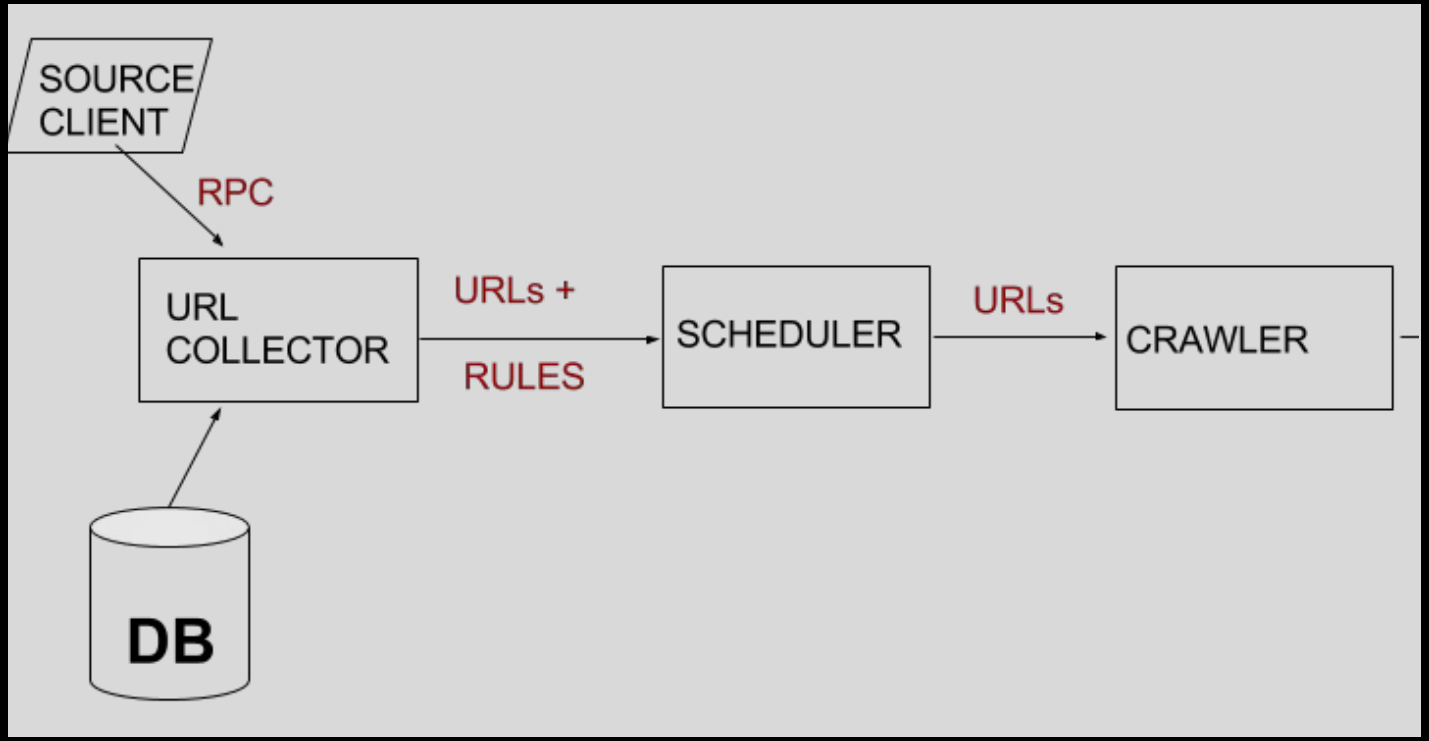
Search Indexing Pipelines



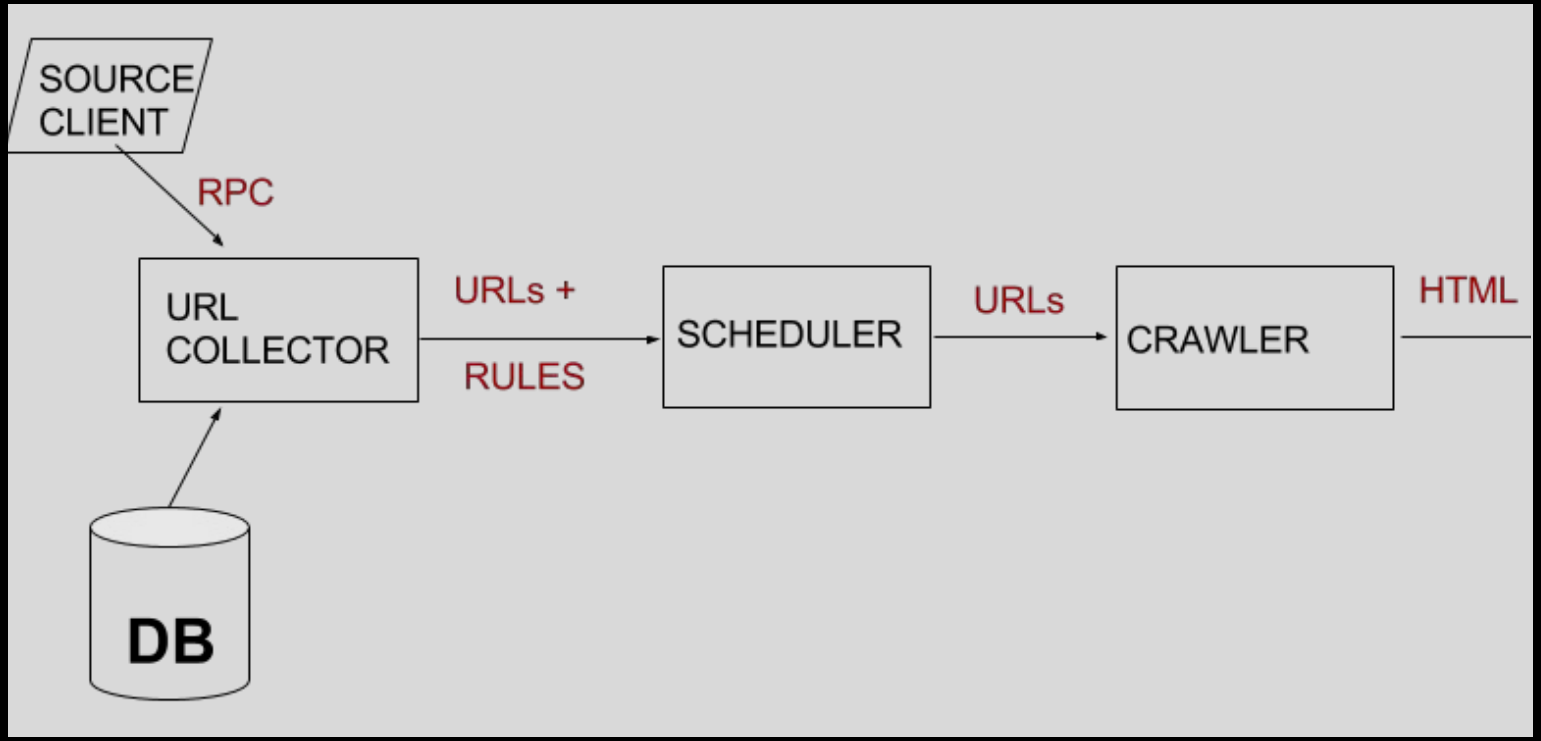
Search Indexing Pipelines



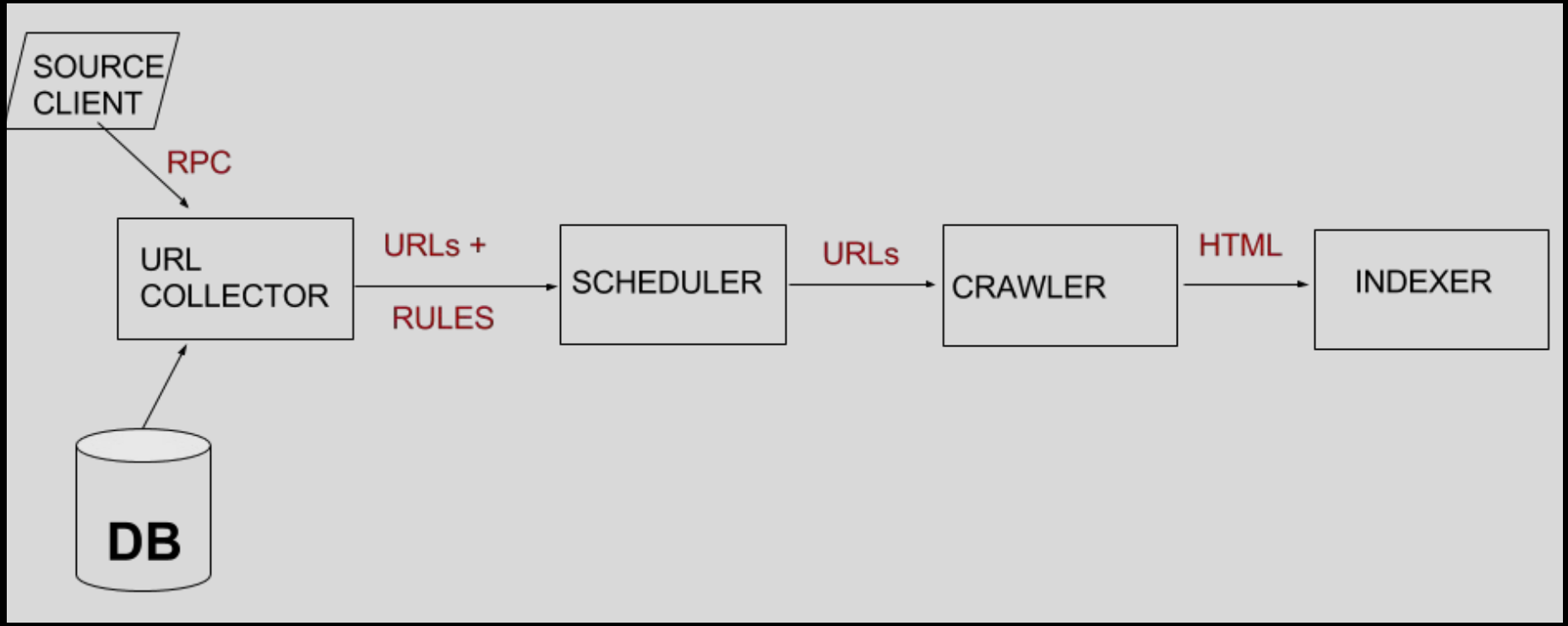
Search Indexing Pipelines



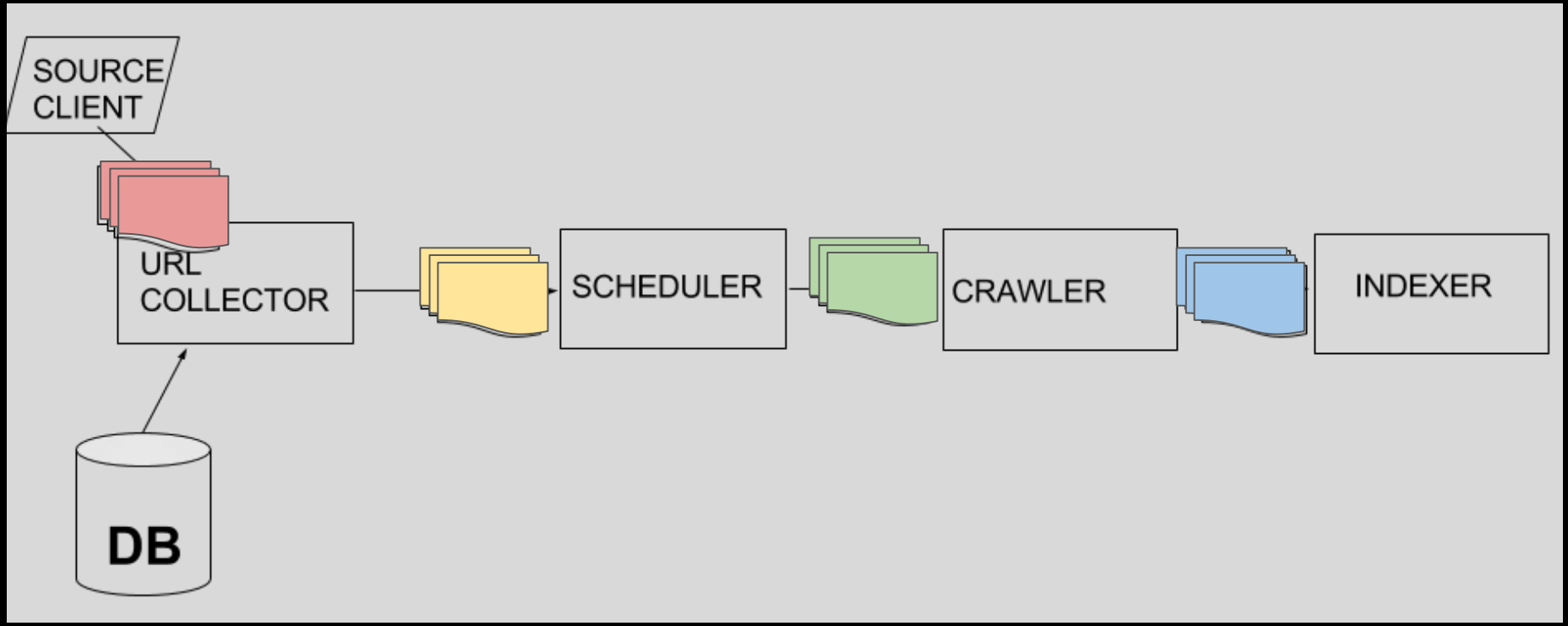
Search Indexing Pipelines



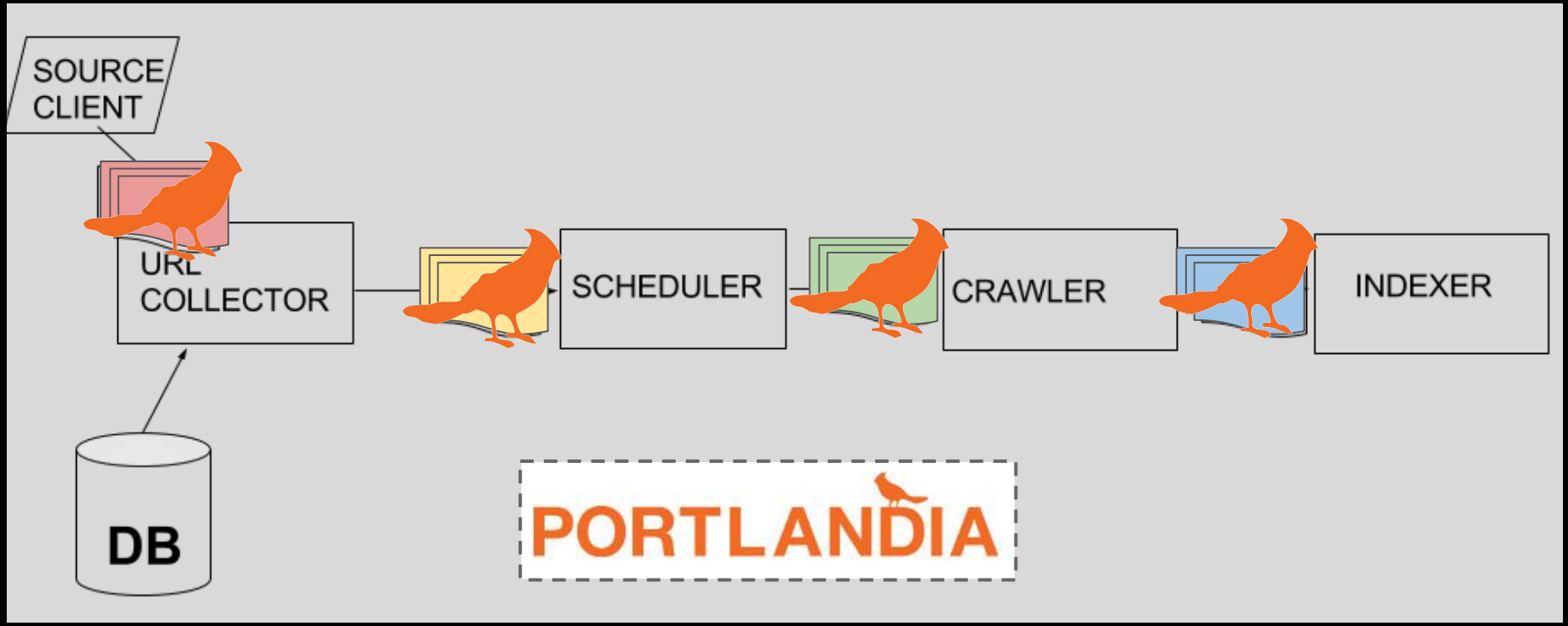
Search Indexing Pipelines



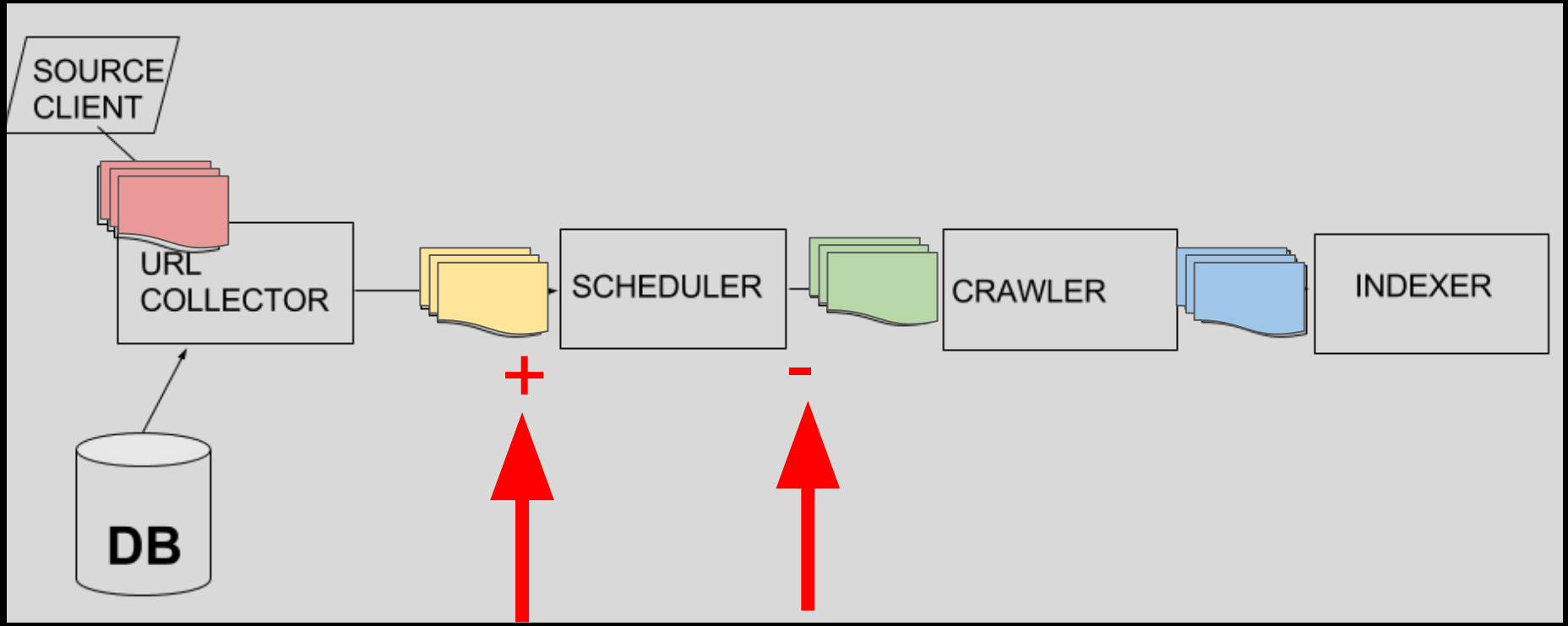
Search Indexing Pipelines



Search Indexing Pipelines



Search Indexing Pipelines



Leases
as Heart Beat in
Distributed
Systems

Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency

Cary G. Gray and David R. Cheriton
Computer Science Department
Stanford University

1989

Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency

Cary G. Gray and David R. Cheriton
Computer Science Department
Stanford University

Leases

- Distributed locking

Leases

- Distributed locking
- Lease term tradeoffs
  - short

Leases

- Distributed locking
- Lease term tradeoffs
  - short vs long

Leases

- Distributed locking
- Lease term tradeoffs
  - short vs long
- Use of leases in modern applications
  - Leader election TTL (in etcd)

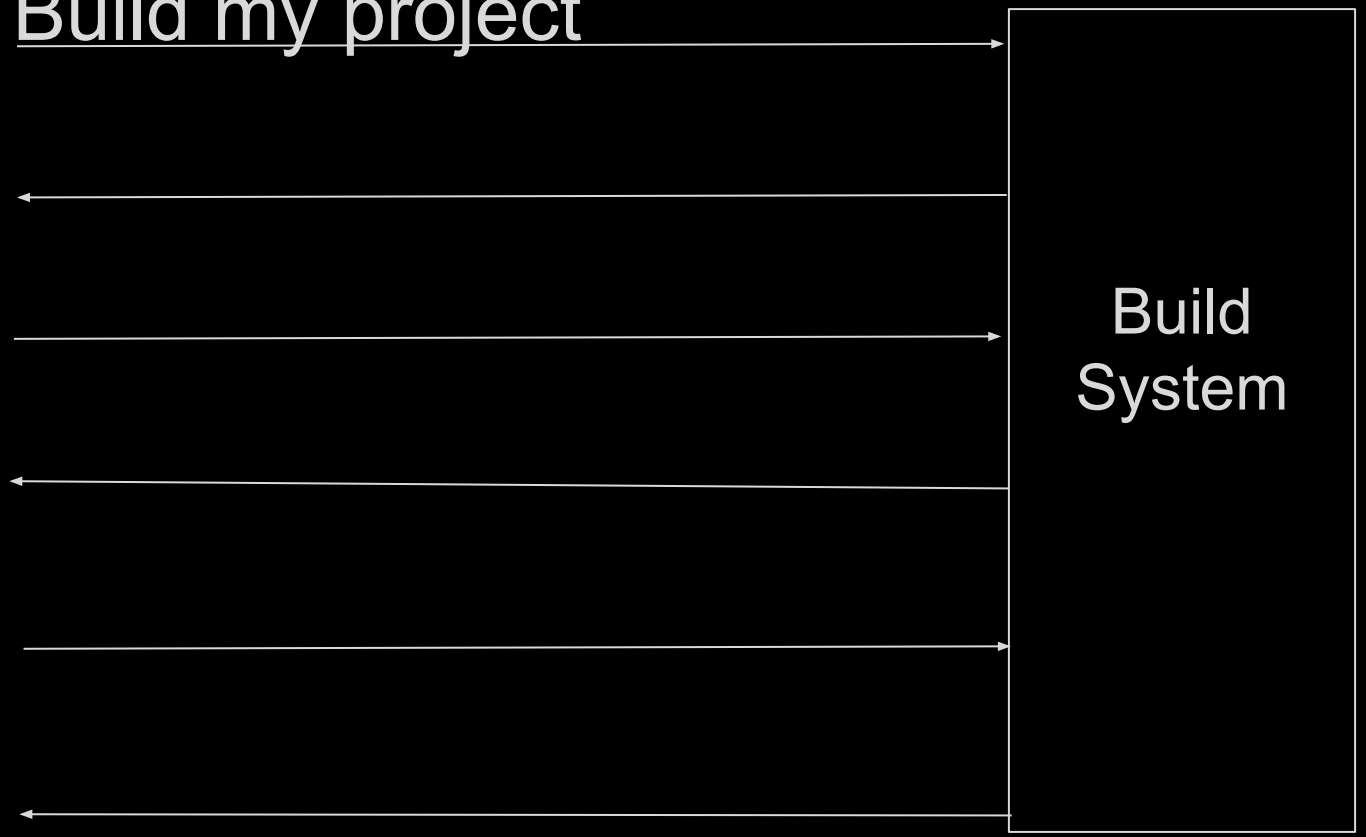
Leases

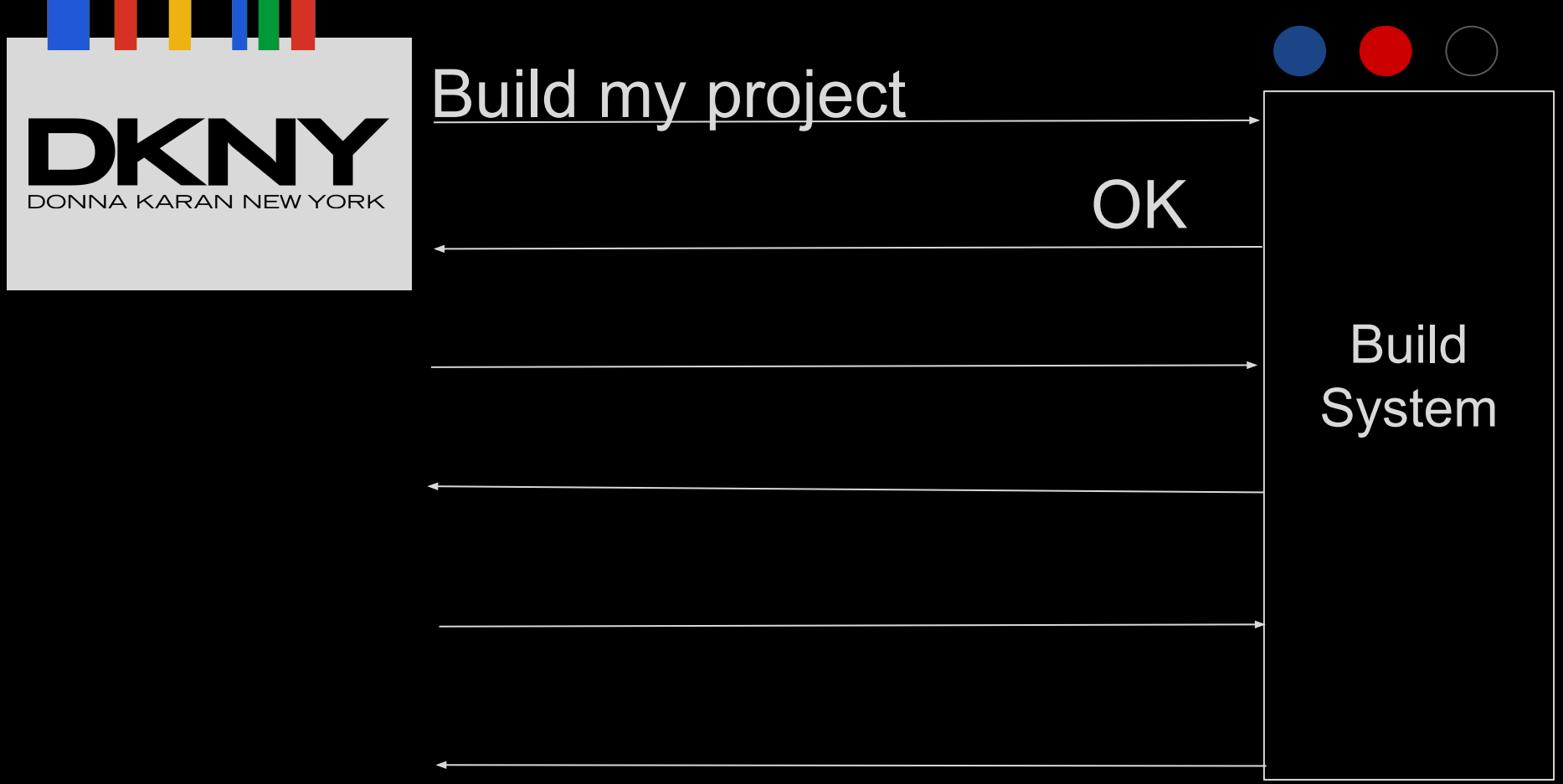
- Distributed locking
- Lease term tradeoffs
  - short vs long
- Use of leases in modern applications
  - Leader election TTL (in etcd)
  - Liveness detection

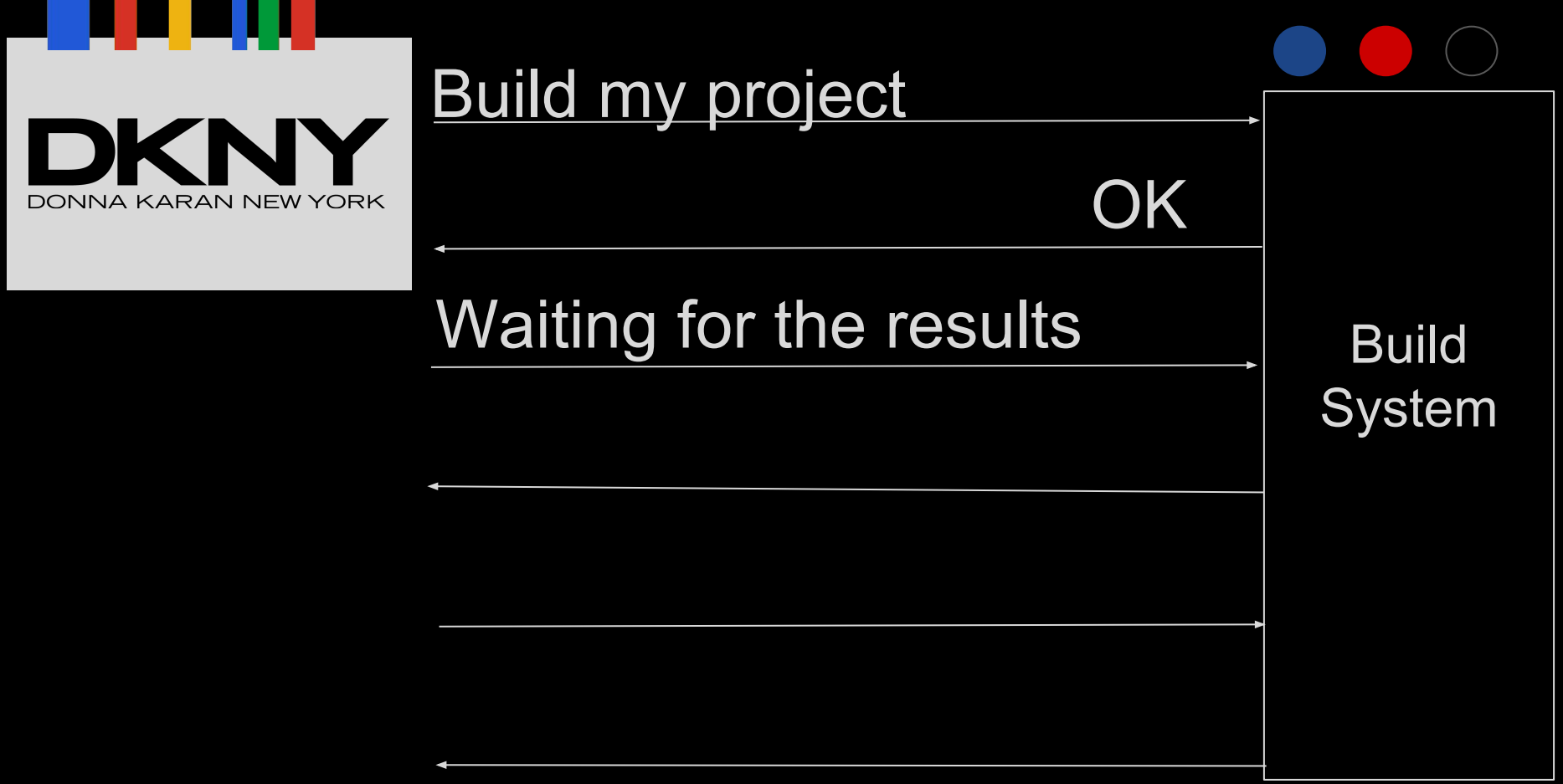

ARE YOU
ALIVE?


Leases in Build System: *Success Scenario*

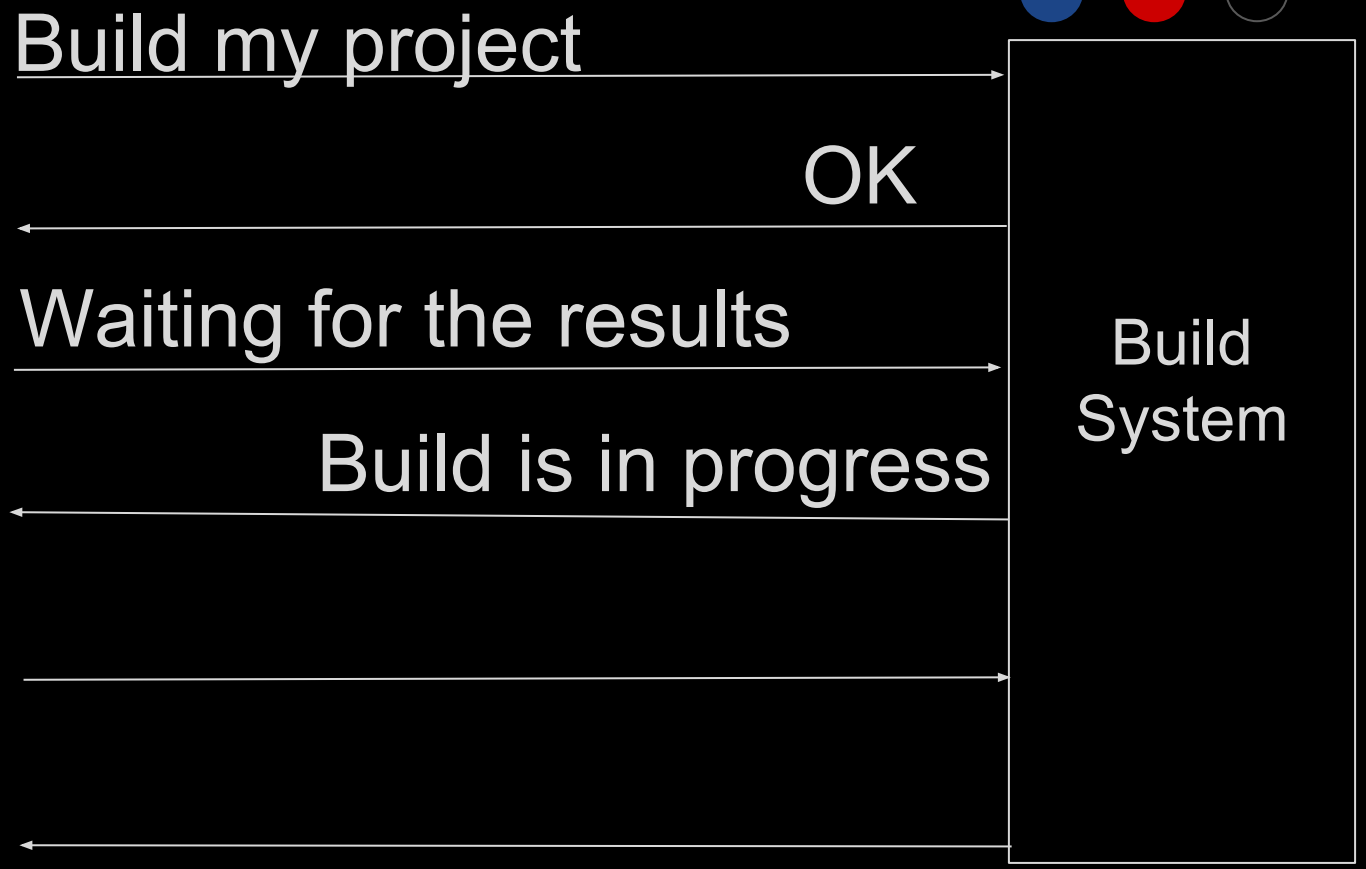
Build my project



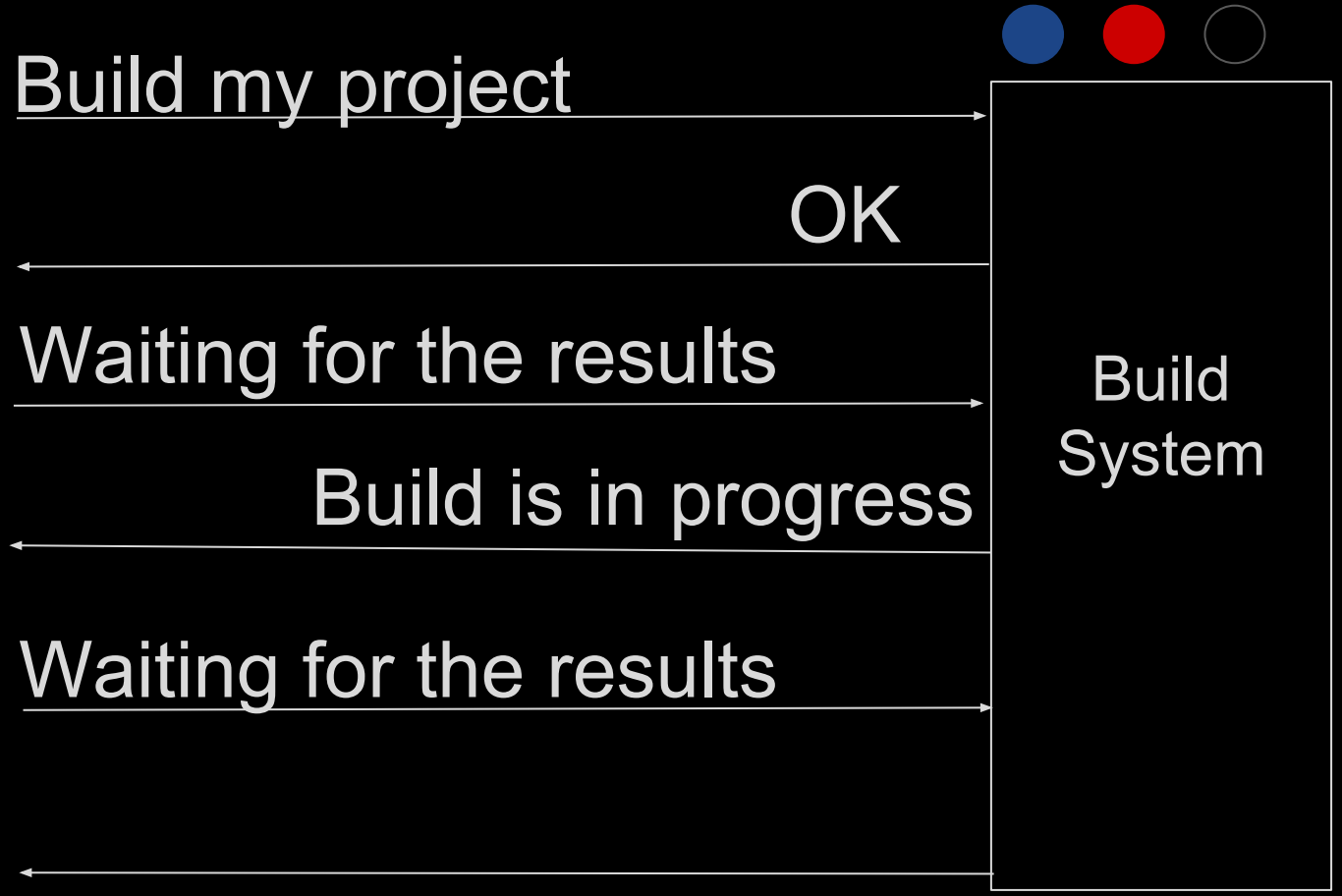

DKNY
DONNA KARAN NEW YORK
Build my project
OK
Build System


DKNY
DONNA KARAN NEW YORK
Build my project
OK
Waiting for the results
Build System


DKNY
DONNA KARAN NEW YORK


Build my project
OK
Waiting for the results
Build is in progress
Build System


DKNY
DONNA KARAN NEW YORK


Build my project
OK
Waiting for the results
Build is in progress
Waiting for the results
Build System

DKNY
DONNA KARAN NEW YORK

Build my project

OK

Waiting for the results

Build is in progress

Waiting for the results

Build is finished

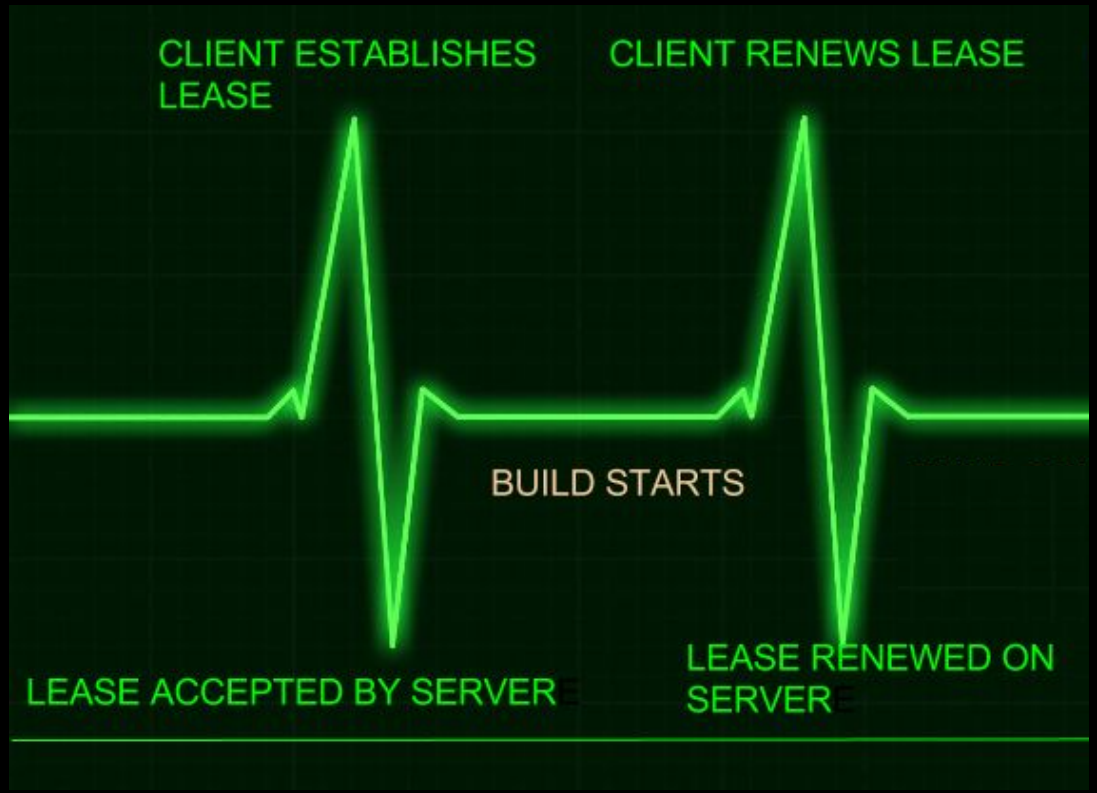
Build System

Leases in Build System: *Failure Scenario*

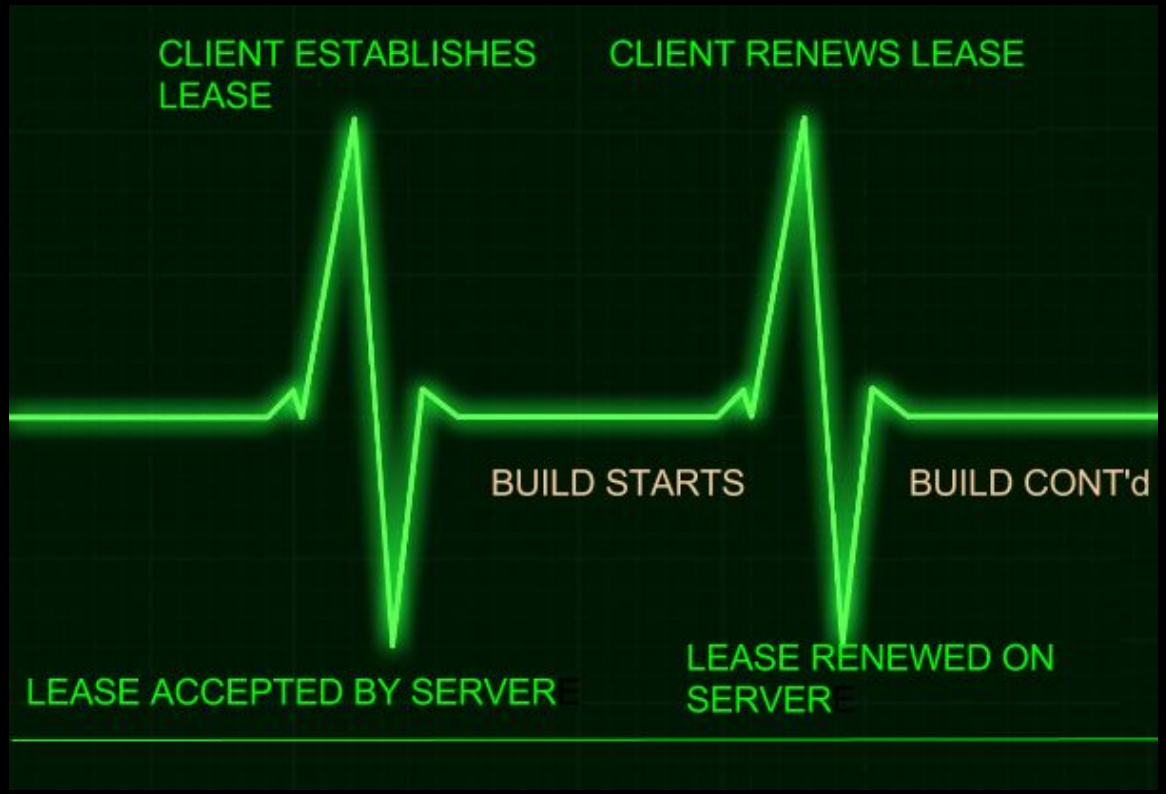
Leases in Build System

Leases in Build System

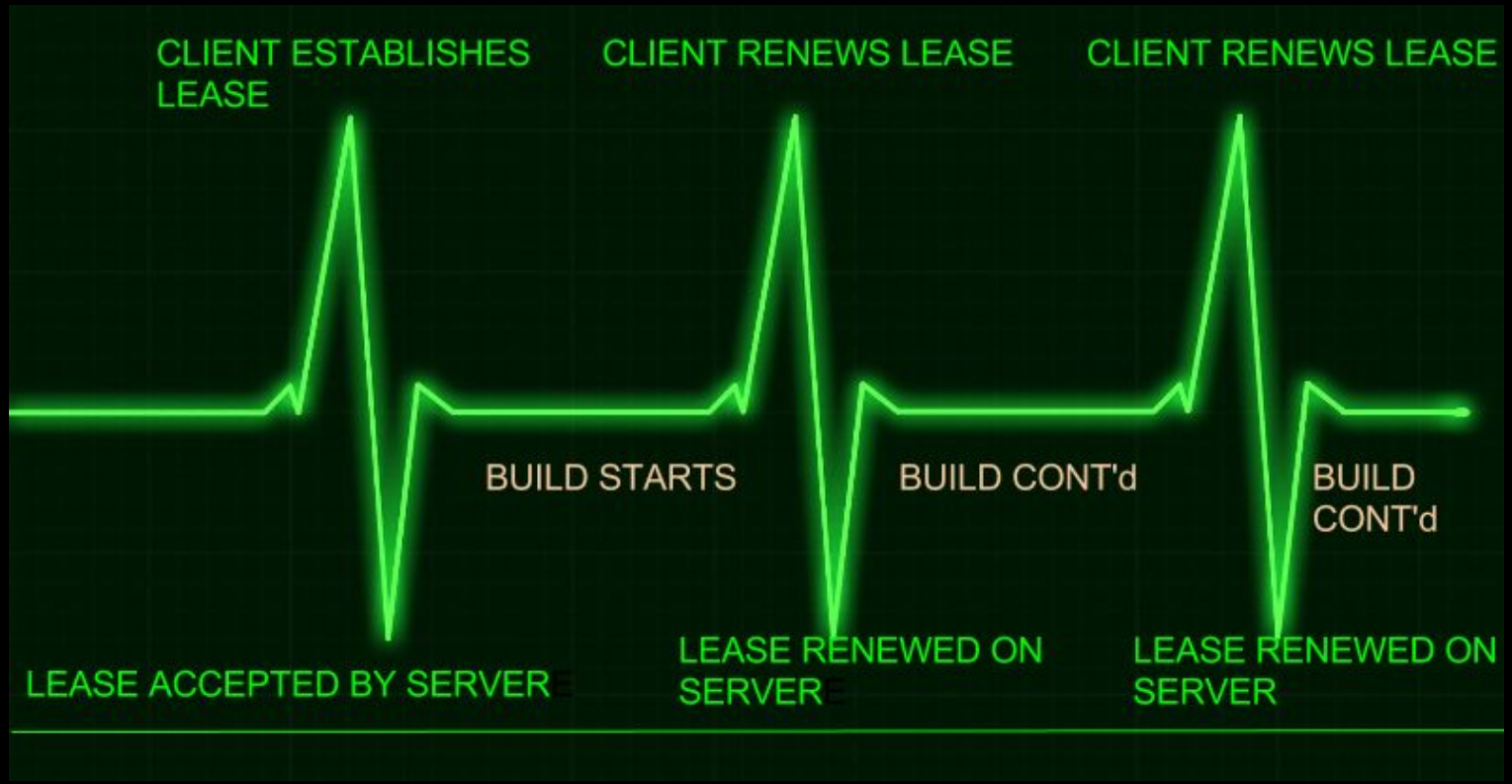
Leases in Build System



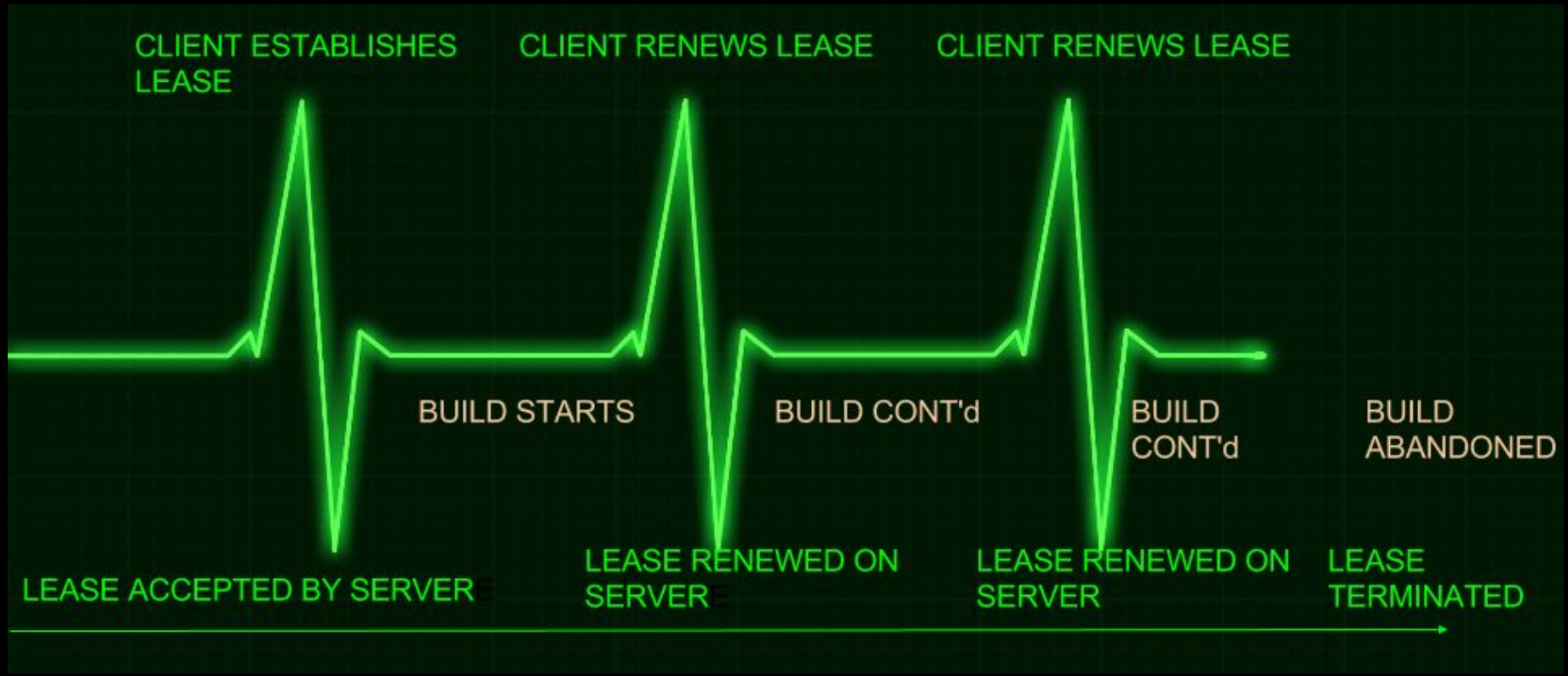
Leases in Build System



Leases in Build System



Leases in Build System



Using etcd leases for heartbeat

```
$ curl http://server.com/v2/keys/foo -XPUT -d\

  value=bar -d ttl=300
```



```
{
    "action": "set",
    "node": {
        "createdIndex": 2,
        "expiration":"2016-06-14T16:15:00",
        "key": "/foo",
        "modifiedIndex": 2,
        "ttl": 300,
        "value": "bar"
    }
}
```

Using etcd leases for heartbeat

```
$ curl http://server.com/v2/keys/foo -XPUT -d \

  value=bar -d ttl=300
```

... 3 minutes later...

Using etcd leases for heartbeat

```
$ curl http://server.com/v2/keys/foo -XPUT -d \
  value=bar -d ttl=300
$ curl \
  http://server.com/v2/keys/foo?prevValue=bar \
  -XPUT -d ttl=300 -d refresh=true -d \
  prevExist=true
```



```
{
    "action": "update",
    "node": {
        "createdIndex": 2,
        "expiration":"2016-06-14T16:18:00",
        "key": "/foo",
        "modifiedIndex": 3,
        "ttl": 300,
        "value": "bar"
    }
    "prevNode": {...}
}
```


```
{
    "action": "update",
    "node"
        "c
        "e
        "k
        "m
        "t
        "v
    }
    "prevN
}
```

```
"prevNode": {
    "createdIndex": 2,
    "expiration":"2016-06-14T16:15:00",
    "key": "/foo",
    "modifiedIndex": 2,
    "ttl": 120,
    "value": "bar"
}
```


Leases for heartbeat:
How long should the lease term be?


Inaccurate Computations
&
Serving Search Results

From Accurate to "Good Enough"

[Trade off] Inaccuracy for Performance

BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data

Sameer Agarwal[†], Barzan Mozafari[°], Aurojit Panda[†], Henry Milner[†], Samuel Madden[°], Ion Stoica[*†]

```
SELECT COUNT(*)
FROM Sessions
WHERE Genre = 'western'
GROUP BY OS
WITHIN 5 SECONDS
```

```
SELECT COUNT(*)
FROM Sessions
WHERE Genre = 'western'
GROUP BY OS
WITHIN 5 SECONDS
```

```
SELECT COUNT(*)
FROM Sessions
WHERE Genre = 'western'
GROUP BY OS
ERROR WITHIN 10% AT CONFIDENCE 95%
```

[Trade off] Inaccuracy for Resilience

Probabilistic Accuracy Bounds for Fault-Tolerant Computations that Discard Tasks *

Martin Rinard
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
rinard@csail.mit.edu


Campbell's
CONDENSED

Reduce

Map

Map

Map

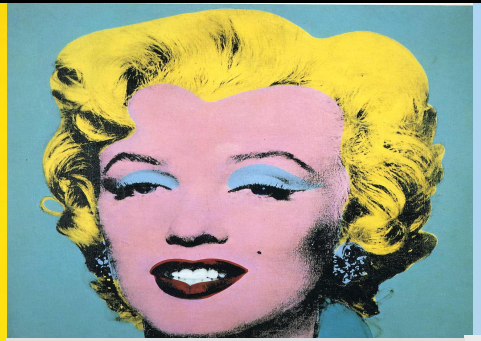
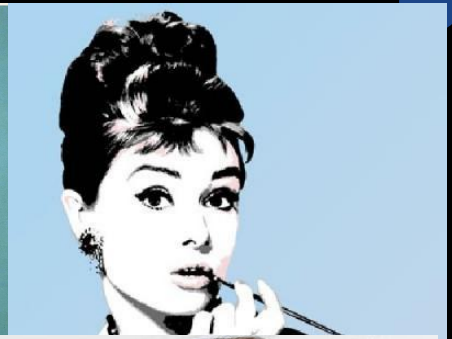
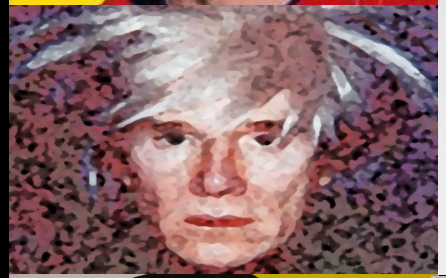
Input

Input

Input

Inaccuracy for Resilience

1. Task decomposition

Campbell's
CONDENSED

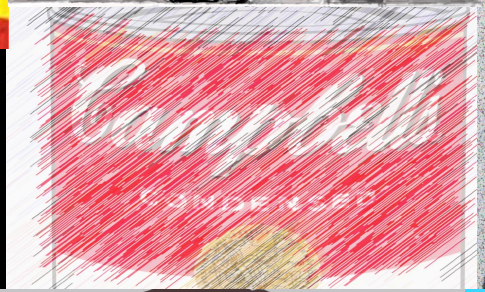


Inaccuracy for Resilience

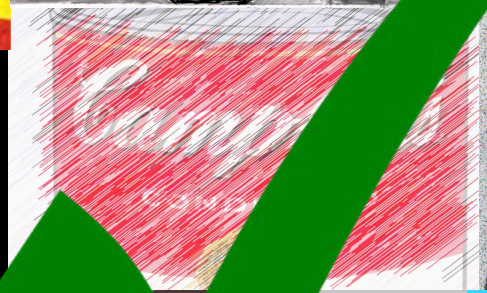
1. Task decomposition
2. Baseline for correctness

Inaccuracy for Resilience

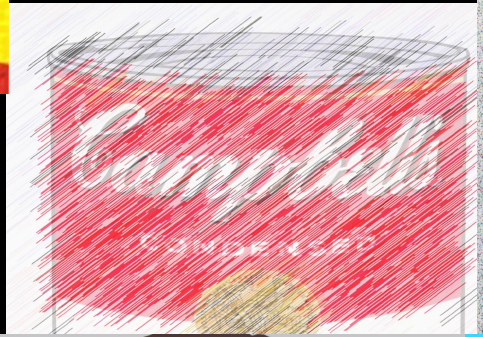
1. Task decomposition
2. Baseline for correctness
3. Criticality Testing

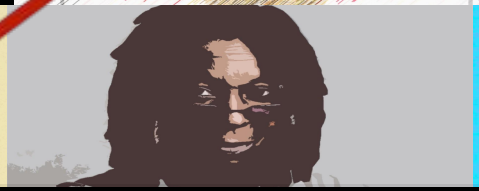









Campbell's



FAILED CRITICALITY TEST

Inaccuracy for Resilience

1. Task decomposition
2. Baseline for correctness
3. Criticality Testing
4. Distortion and timing models

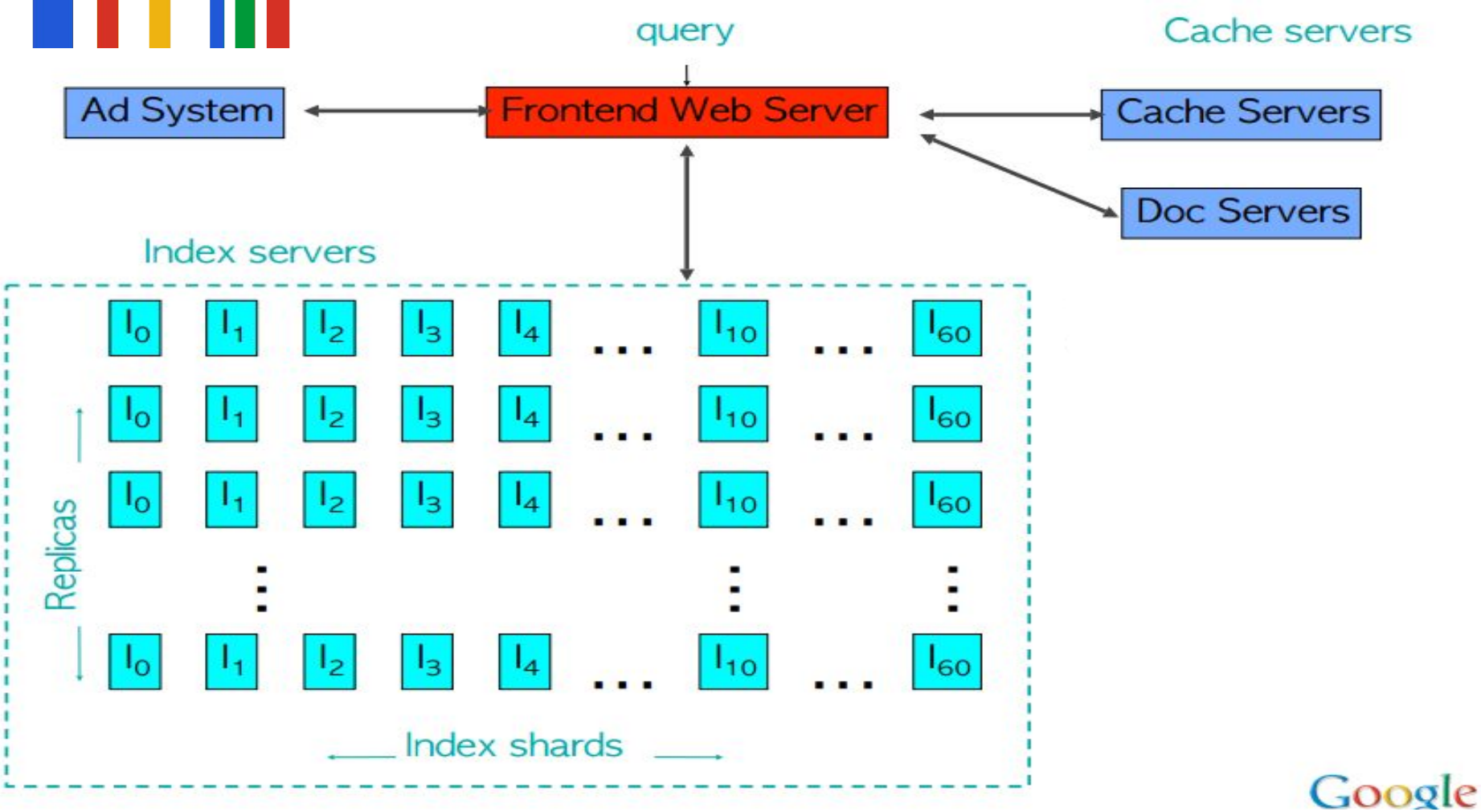
Distortion Model

$$\hat{d}_{1000}(x_1, \ldots, x_4) = 0.010[\pm 0.0012] + 0.053[\pm 0.0066]x_1 + 0.11[\pm 0.0067]x_2 + 0.56[\pm 0.0067]x_3 + 0.54[\pm 0.0067]x_4$$

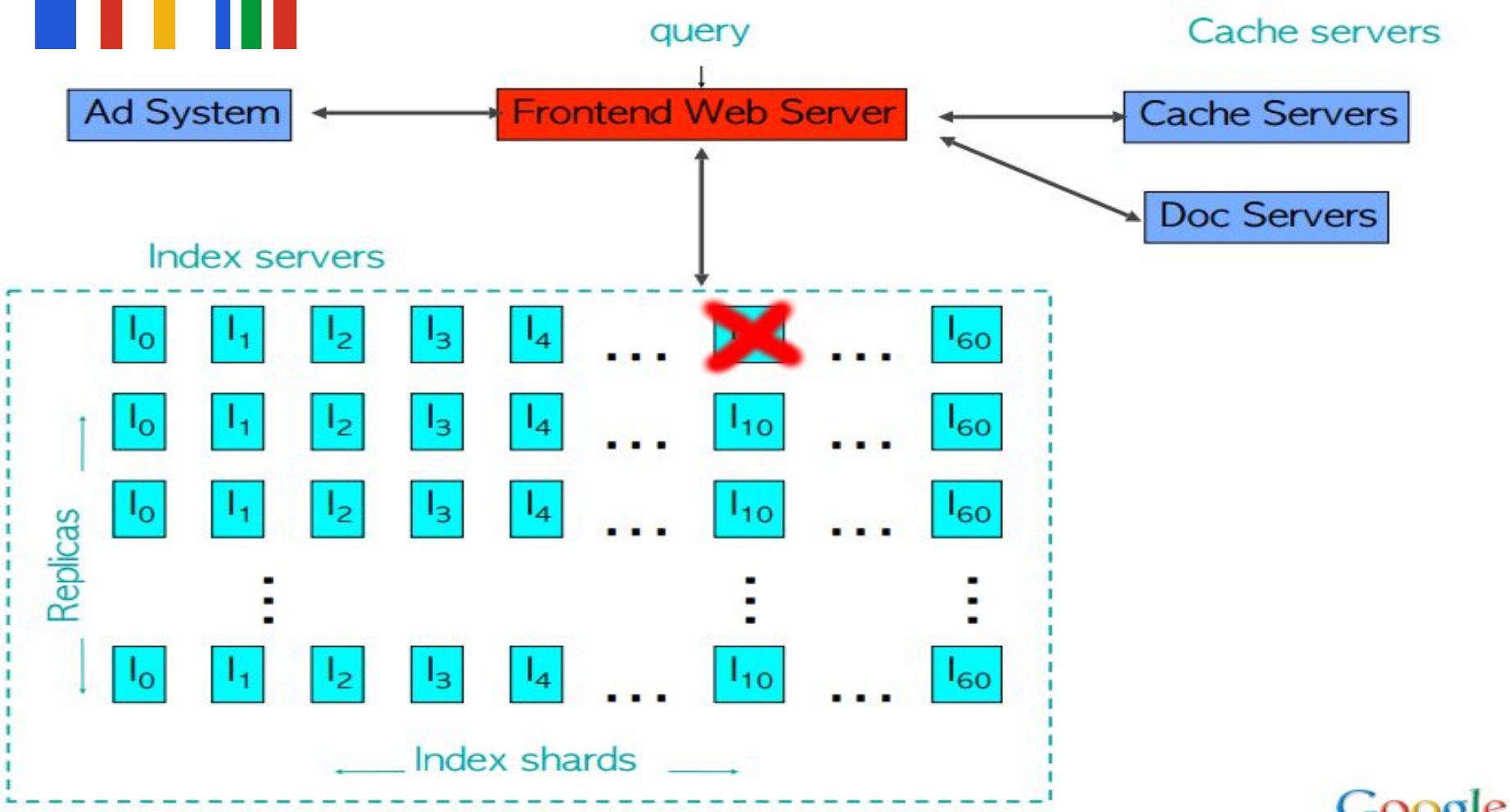
Timing Model

$$\hat{s}_4(x_1, x_2) = 1.0[\pm 0.0007] + -0.87[\pm 0.004]x_1 + -0.047[\pm 0.003]x_2$$

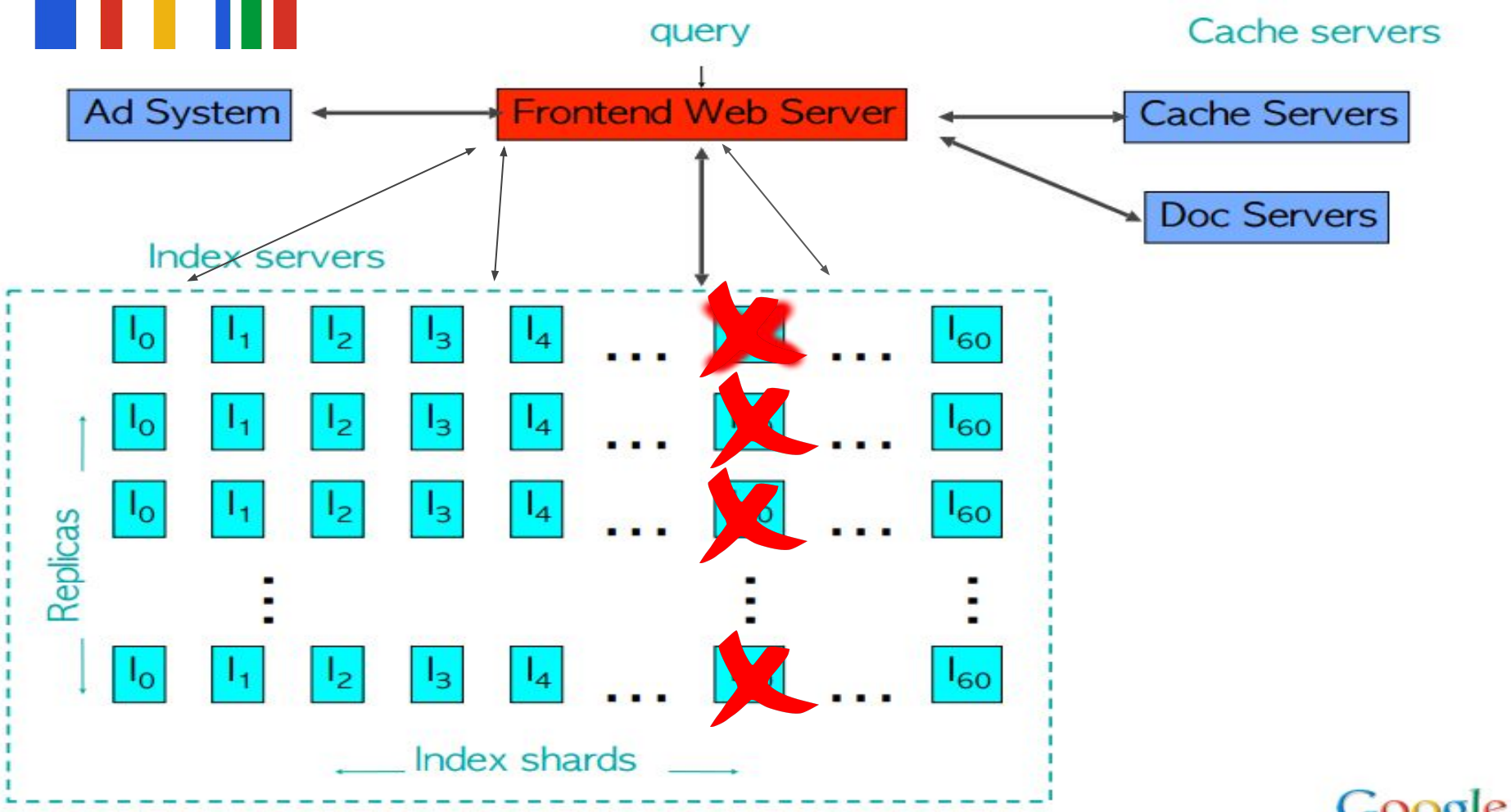
[In production]
Inaccuracy for Performance & Resilience



Jeff Dean "Building Software Systems at Google and Lessons Learned", Stanford, 2010


query
Cache servers
Ad System
Frontend Web Server
Cache Servers
Doc Servers
Index servers
I_0 I_1 I_2 I_3 I_4 ... ... I_{60}
I_0 I_1 I_2 I_3 I_4 ... I_{10} ... I_{60}
I_0 I_1 I_2 I_3 I_4 ... I_{10} ... I_{60}
I_0 I_1 I_2 I_3 I_4 ... I_{10} ... I_{60}
Replicas
Index shards

Google


query
Cache servers
Ad System
Frontend Web Server
Cache Servers
Doc Servers
Index servers
Replicas
Index shards
Google


[Designing with]
Inaccuracy for Performance & Resilience


[Designing with]
Inaccuracy for Performance & Resilience

applicable to some problem domains

focus on observability

simplified implementation

fuzz testing

fault injection testing

generative testing

[Designing with]
Inaccuracy for Performance & Resilience

applicable to some problem domains

focus on observability

simplified implementation

References

- T. Wurthinger, C. Wimmer et al. "One VM to Rule Them All"
- M. Rinard "Probabilistic Accuracy Bounds for Fault-Tolerant Computations that Discard Tasks"
- F. Corbato, M. Daggett, R. Daley "An Experimental Time-Sharing System"
- E. Dijkstra "Cooperating Sequential Processes"
- L. Lamport "Time, Clocks, and the Ordering of Events in a Distributed System"
- http://blinkdb.org/


References

- B. Oki, B. Liskov "Viewstamped Replication: A New Primary Copy Method to Support Highly-Available Distributed Systems"
- L. Lamport "The Part-Time Parliament"
- M. Welsh, D. Culler, E. Brewer "SEDA: An Architecture for Well-Conditioned, Scalable Internet Services"
- C. Gray, D. Cheriton "Leases: An Efficient Fault-Tolerant Mechanism for Distributed File Cache Consistency"
- S. Agarwal, B. Mozafari et al. "BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data"

Gratitude

Ines Sombra

David Greenberg

Karan Parikh

Matt Welsh

Erran Berger


Robust & scalable pipelines


Robust & scalable pipelines

Leases for sharing & heartbeat

Robust & scalable pipelines

Leases for sharing & heartbeat

Inaccuracy for resilience & performance


Robust & scalable pipelines

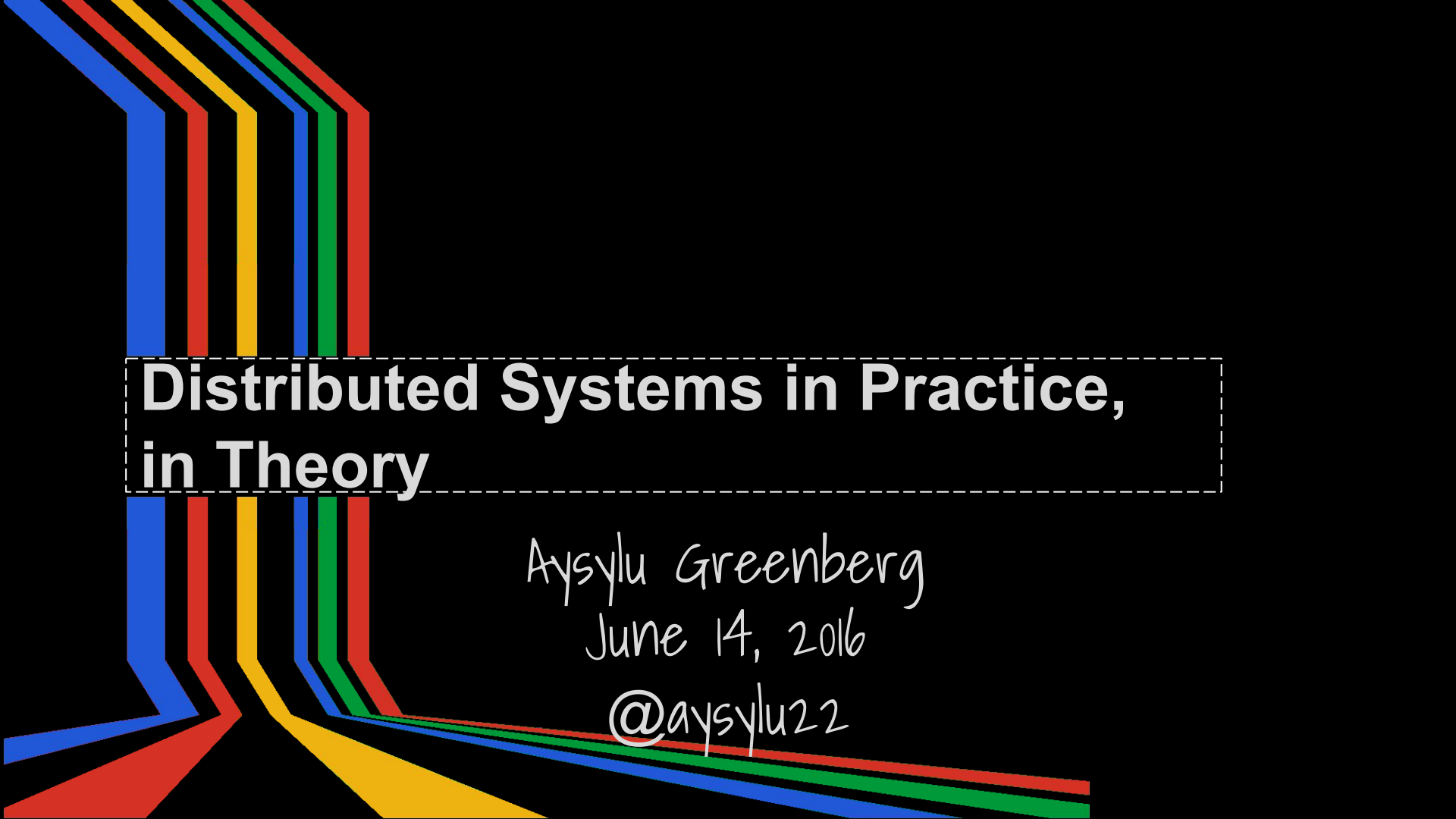
Leases for sharing &

heartbeat

Inaccuracy for resilience &

performance

CS research is timeless: use it to mitigate risk


Distributed Systems in Practice, in Theory
Aysylu Greenberg
June 14, 2016
@aysylu22