Cache, Workers, and Queues

JMac

@gonedark

Tools for your application at scale

JMac != expert

willcoxcorvette.local

Inbox - jason@pureconcepts.net - Pur...
localhost / 127.0.0.1 / willcoxc_db_wc...
Index Management / System / Mage...
Stack Overflow
http timeout - Google Search

The connection has timed out

The server at [redacted] is taking too long to respond.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

Try Again

“Increasing the timeout is not a solution.”

– @gonedark

Things that don't scale

- **Repeated, Expensive Operations:** queries, calculations, etc…
- **Ancillary Operations:** sending email, processing uploads, etc…
- **Notifications**

Optimization

“At some point, optimization doesn’t scale.”

#1 Repeated, Expensive Operations

Queries

“The average WordPress page executes 35 queries.”

"So, what do we do?"

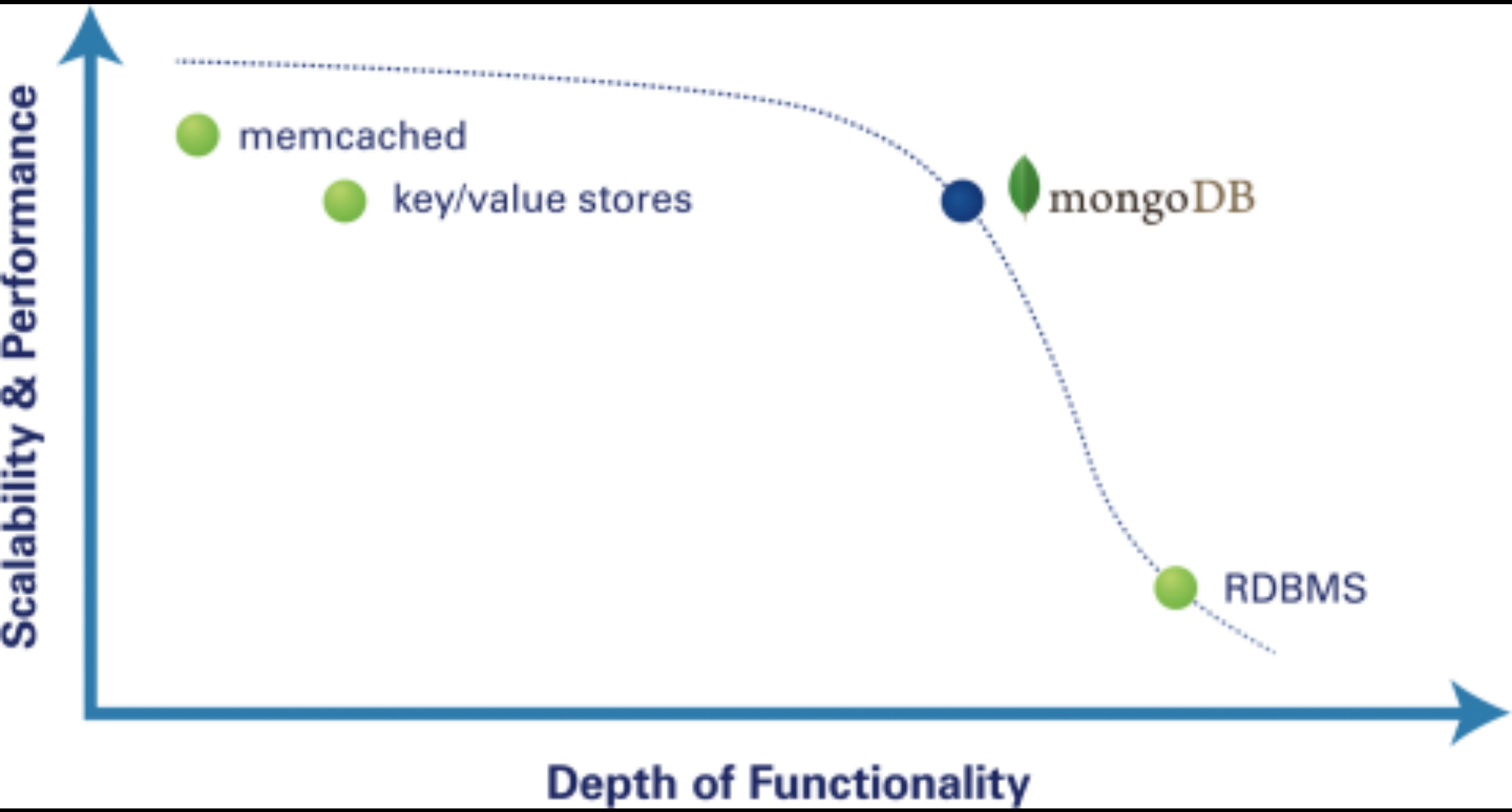
– The Audience

CACHE

ALL THE THINGS!

Cache?

Cache == Datastore


Scalability & Performance
memcached
key/value stores
mongoDB
RDBMS
Depth of Functionality

Caches

- memcached
- redis
- NCache
- Amazon ElastiCache
- Azure Shared Cache

```
class User {
    public getAllUsers() {
        sql = 'SELECT * FROM users'
        users = datastore.query(sql)

        return users
    }
}
```

```
class User {
    public getAllUsers() {
        users = cache.get('alluserkey')
        if (users) {
            return users;
        }

        sql = 'SELECT * FROM users'
        users = datastore.query(sql)

        cache.set('alluserkey', users)

        return users
    }
}
```

```
class UserCache {
    public getAllUsers() {
        users = cache.get('alluserkey')
        if (users) {
            return users;
        }
        users = super.getAllUsers()
        cache.set('alluserkey', users)

        return users
    }
}
```

Common Architectures

- **together:** better utilization of resources, but individual and smaller caches
- **standalone:** larger, shared cache, but requires additional resources

Considerations

- data size limits per key
- expiration
- stale data
- replication/consistency

#2 Ancillary Operations

Examples

- sending email
- processing images
- recalculations
- system interactions

"So, what do we do?"

– The Audience

Workers

Asynchronous Processes

Workers

- Gearman
- Beanstalkd
- Amazon Simple Queue
- Azure Webjobs?
- You can create a job queue with a datastore…

```
handleProfileImageUpload(file) {
    moveFile(file, PROFILE_IMAGE_DIR)
    createThumbnails(file)
}
```

```
handleProfileImageUpload(file) {
    moveFile(file, PROFILE_IMAGE_DIR)
    createThumbnails(file)
}
```

```
handleProfileImageUpload(file) {
    moveFile(file, PROFILE_IMAGE_DIR)
    JobQueue.push(
        'createThumbnails',
        $file
    )
}
```

```
class ProcessImages extends Job {
    public createThumbnails($file) {
        // do work
    }
}
```

Common Architectures

- **together:** shared resources, but might be risky resource intensive jobs
- **standalone:** separate resources, but requires additional resources

Considerations

- complexity
  - asynchronicity
  - idempotency
  - failure
- additional resources
- environmental difference

#2 Notifications

Examples

- system
- in-app
- user
- third-party

"So, what do we do?"

– The Audience

Message Queues

Pub/Sub

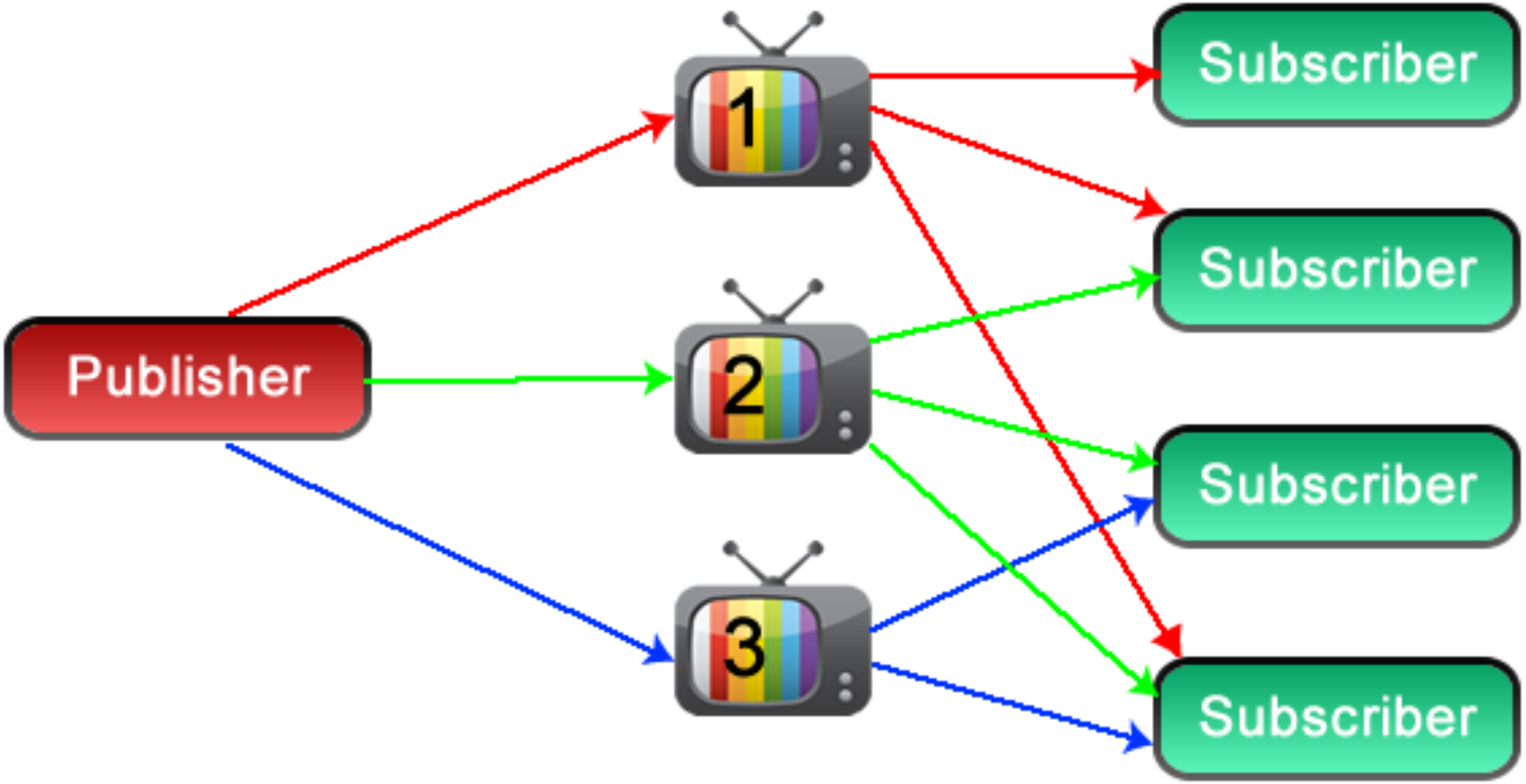
WAT.

Publish/Subscribe

Broadcasting

Messaging Queues

- RabbitMQ
- ActiveMQ
- *MQ
- redis

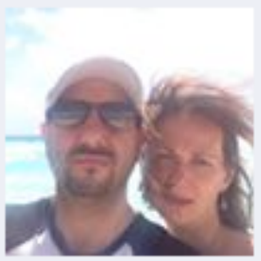

Publisher
1
2
3
Subscriber
Subscriber
Subscriber
Subscriber

Notifications

Mark as Read · Settings

Kim Lilly Horine, **Xal Glover** and 3 other people like your photo.

45 minutes ago



Daniel Straus commented on your photo.

about an hour ago

Today is **David Thomas**'s birthday.

4 hours ago

Jeff Krawiec added a new photo.

14 hours ago

```
getNotifications()
{
  notifications = []

  // [‘user134.*’, ‘user567.birthday’]
  foreach(user.channels as channel) {
    messages = MQ.read(channel)

    if (messages) {
      notifications.push(messages)
    }
  }

  return notifications
}
```

Common Architectures

- **together:** shared resources, but might be a single point of failure
- **standalone:** redundancy, but requires additional resources

Considerations

- additional resources, environmental concerns, complexity
- protocols
- data coupling
- message delivery
- “broadcast storms”
- redundancy

Cache, Workers, and Queues

Questions

Thanks!

@gonedark