



Designing to Scale

The Story of ShootQ
Jonathan LaCour - CTO
jonathan@shootq.com

A Bit of History



[PORTFOLIO](#)

[SLIDESHOWS](#)

[ABOUT](#)

[THE EXPERIENCE](#)

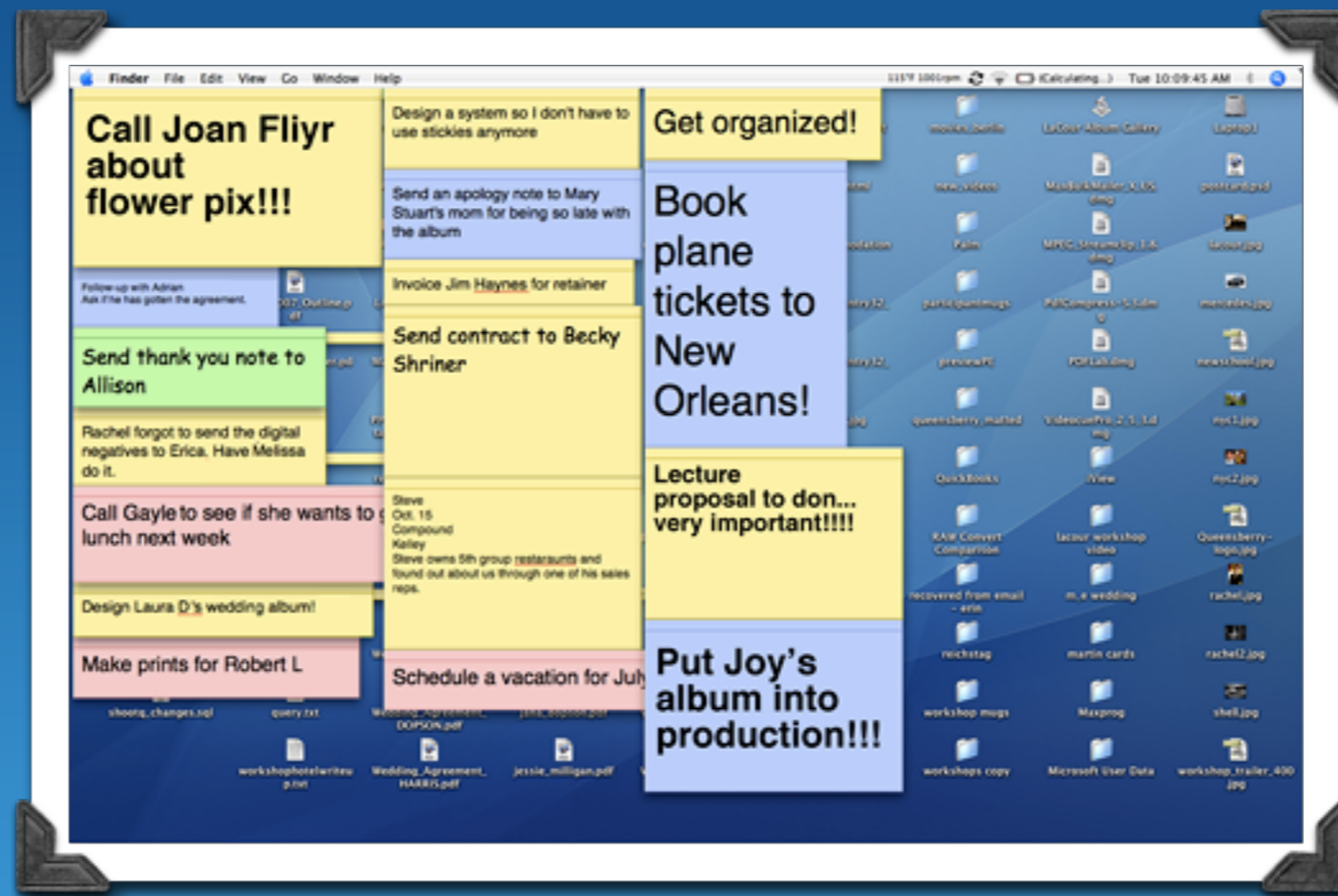
LACOUR

[BLOG](#)

[CLIENT ACCESS](#)

[CONTACT](#)

Organizational Nightmare



Organizational Nightmare

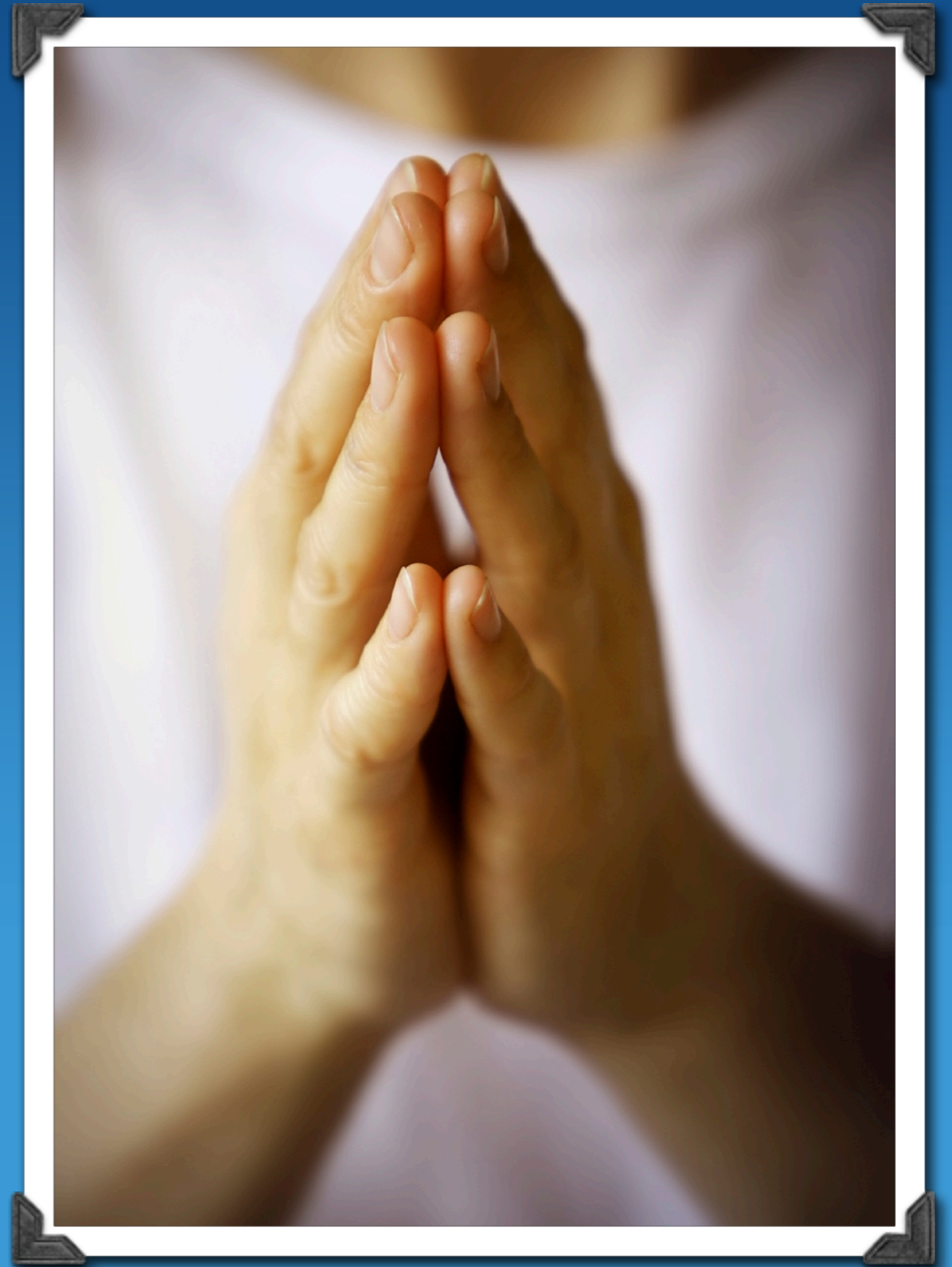
Andrew Begs

Andrew Begs



Andrew Begs

“Can you help me?”



My Response...

My Response...

Buy a book!

My Response...

Buy a book!

THE EXPERT'S VOICE® IN OPEN SOURCE

Beginning PHP and MySQL E-Commerce

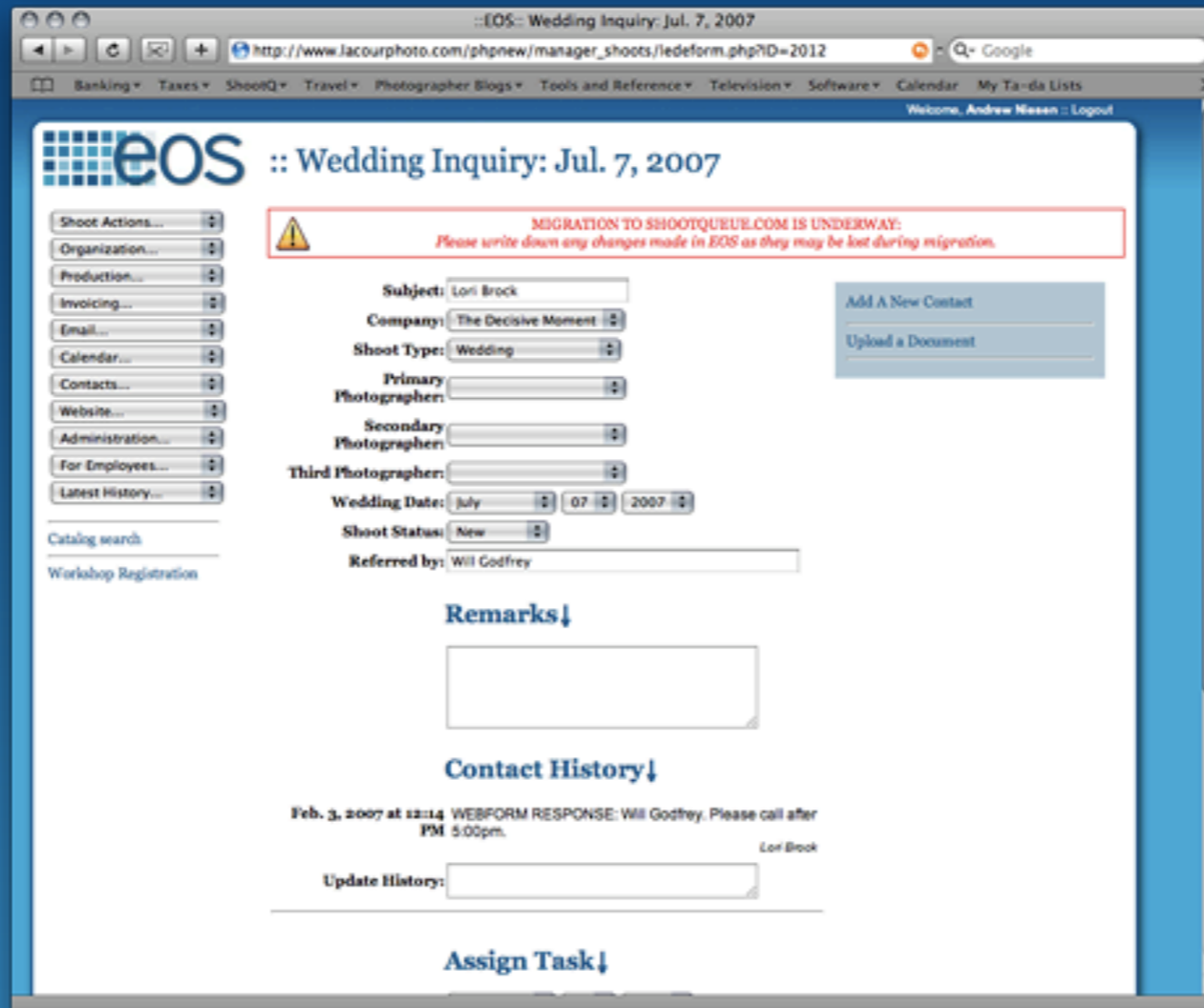
From Novice to Professional

This practical PHP and MySQL tutorial will teach you how to successfully design and build fully featured e-commerce web sites.

SECOND EDITION

Cristian Darie and Emilian Balanescu

Apress®



EOS is Born!

Four years later...

Four years later...

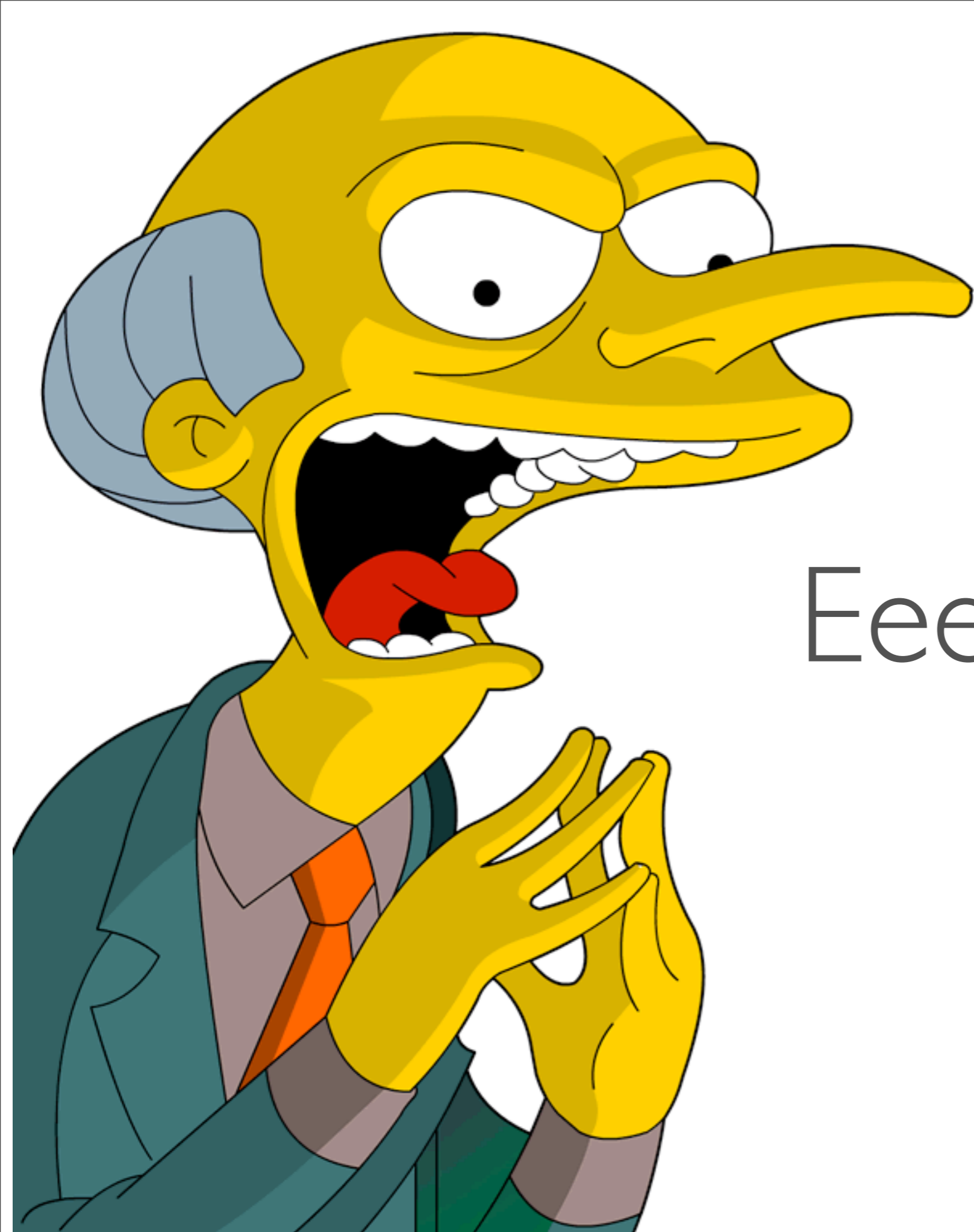
.... other people want to use EOS.

Four years later...

.... other people want to use EOS.

They are even willing to pay for it!





Eeeexcellent.

ONE PROBLEM

ONE PROBLEM

the code is...





Python to the rescue!

Python to the rescue!

Time to start over...

Selecting our Tools

Selecting our Tools

Selecting our Tools

- TurboGears 2.0
 - Object-Dispatch
 - WSGI at the core
 - Genshi and Mako

Selecting our Tools

- TurboGears 2.0
 - Object-Dispatch
 - WSGI at the core
 - Genshi and Mako
- SQLAlchemy / Elixir
 - Easy to use
 - Handles Complex Data
 - Active Community

Scaling Up

What is Scaling?

What is Scaling?

Vertical Scaling

What is Scaling?

Vertical Scaling

Horizontal Scaling

What is Scaling?

Vertical Scaling

- Adding resources
 - CPU, RAM, I/O
- Single server
- Maximizing utilization of resources

Horizontal Scaling

What is Scaling?

Vertical Scaling

- Adding resources
 - CPU, RAM, I/O
- Single server
- Maximizing utilization of resources

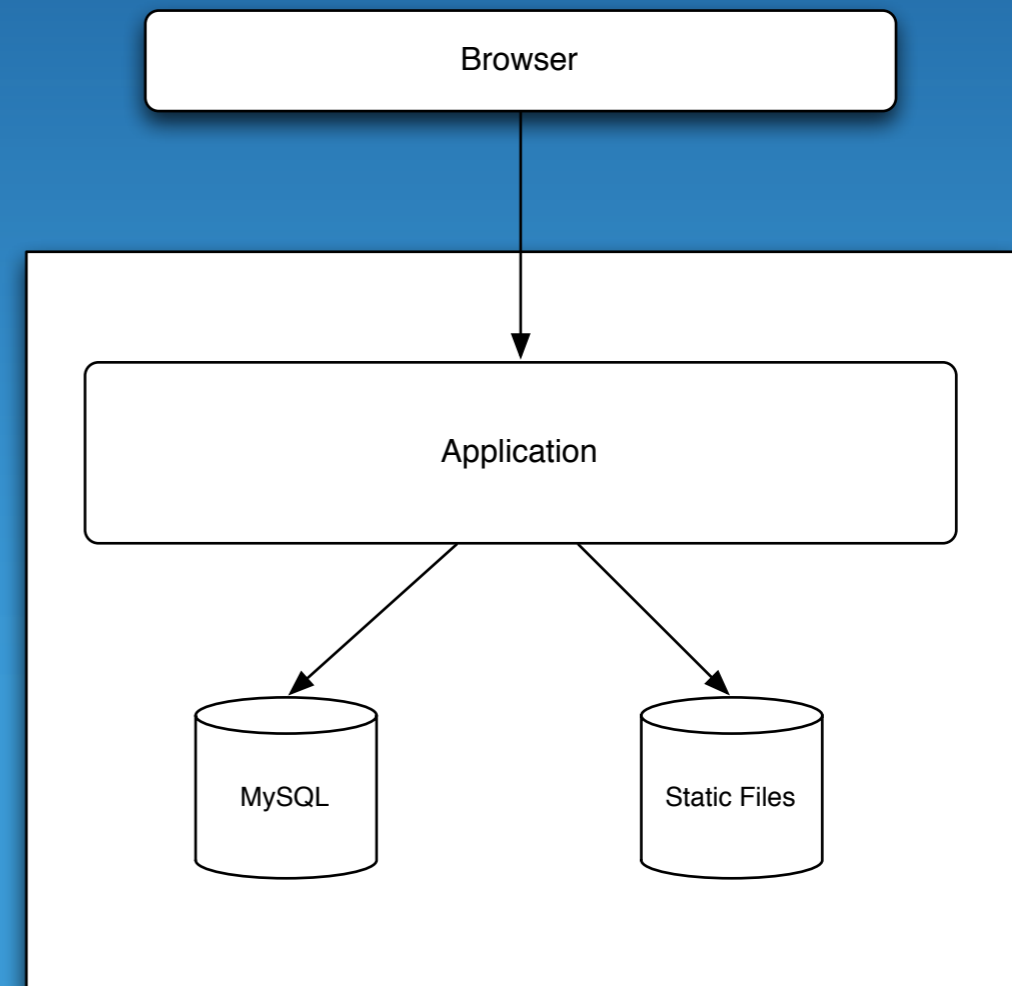
Horizontal Scaling

- Adding servers
- Spreading load
- Separation of concerns
- Limiting resource contention

Vertical Scaling

Naive Infrastructure

- ShootQ 1.0 (PHP) used this infrastructure
- Single server
- Application in charge
 - Talks to database
 - Serves static files



Problem: Static Files

Static Files

Static Files

- App is for dynamic content, not static content

Static Files

- App is for dynamic content, not static content
- Web servers are designed for this job

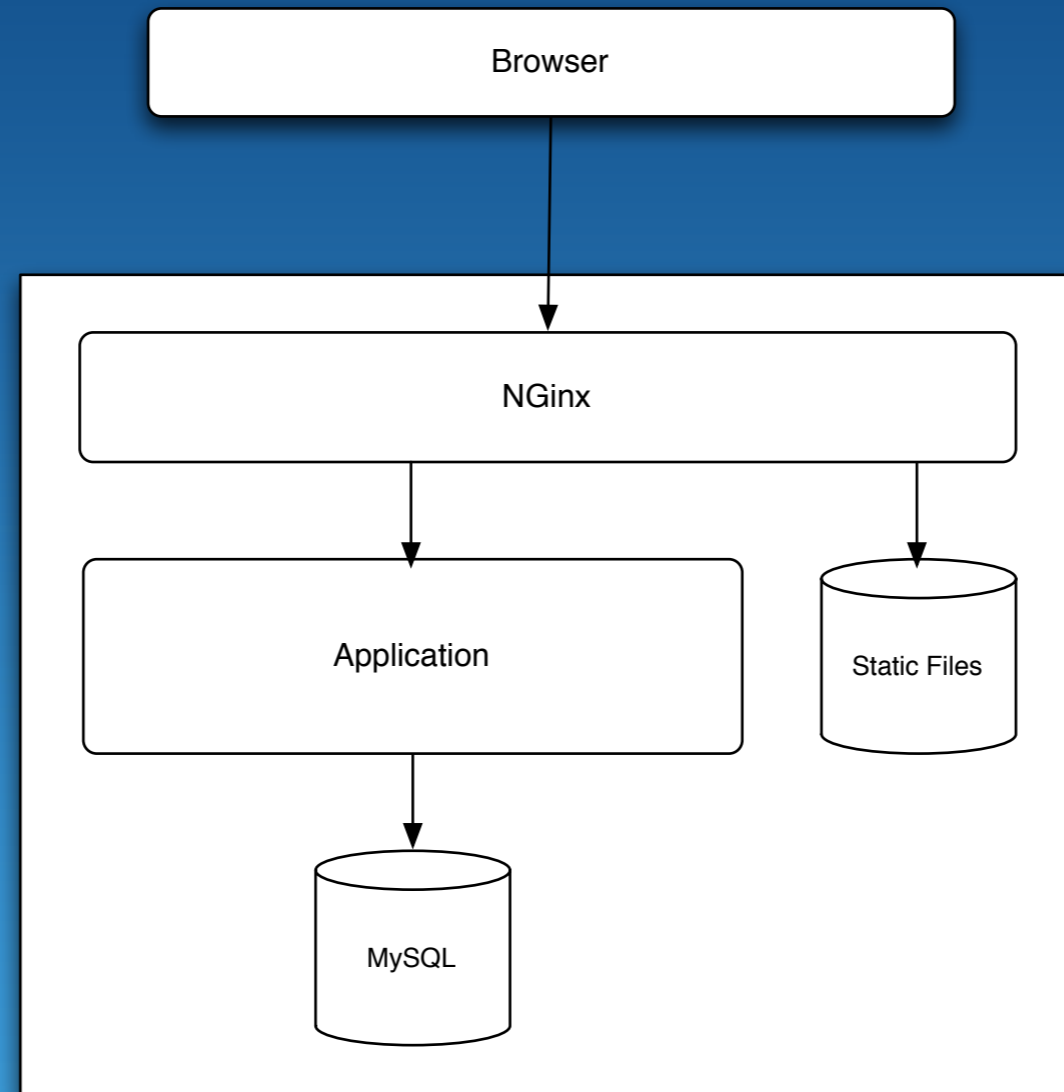
Static Files

- App is for dynamic content, not static content
- Web servers are designed for this job
- Options abound!
 - Apache
 - Lighttpd
 - Many many more...





“engine ex”



Improved Infrastructure

Side Benefits – Cache and GZip

- Nginx Cache Headers
 - Force sixty day cache
- We add a “stamp” to URIs
 - Forces fetch when files are updated
- Content is also GZip'd

```
def resource(path):  
    return '%s?%s' % (  
        path,  
        config['stamp']  
    )
```

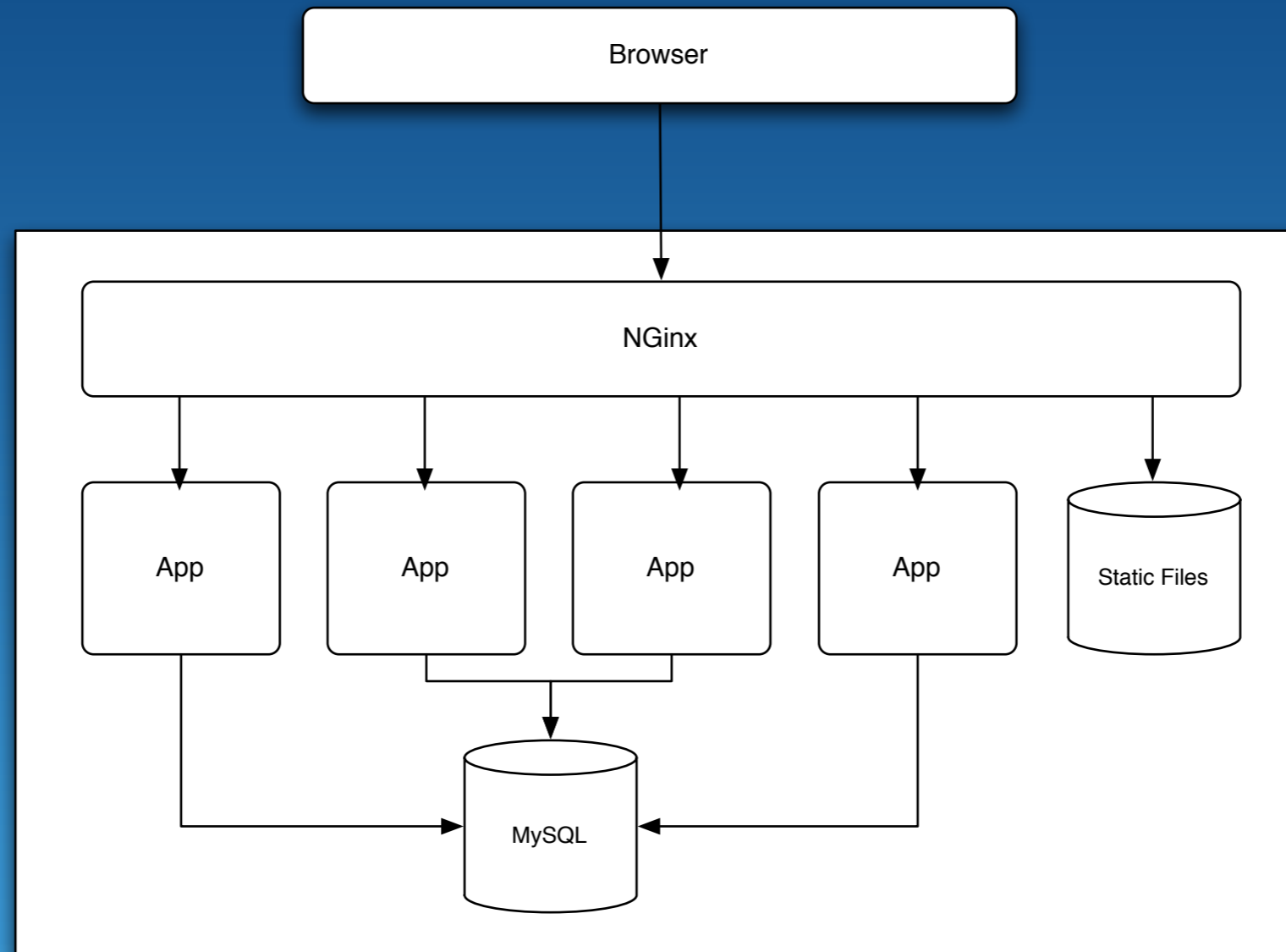
Problem: Many Requests

Threaded WSGI Server

- ShootQ runs inside CherryPy WSGI Server
- Threaded server
- Python threads hampered by GIL
- How to take advantage of multiple cores?

NGinx Clusters

- Run multiple instances of your application
- NGinx will proxy to a “cluster” of instances
- Requests divided between instances
- This is essentially load balancing



Clustered Infrastructure

Problem: Adding Resources

Adding Resources

- Adding CPU, Memory, Disk requires a reboot
- Purchasing hardware not in our budget
- Prefer to focus on the software, not the infrastructure

Deploying in the Cloud

- Joyent is a cloud computing provider



Deploying in the Cloud

- Joyent is a cloud computing provider
- Virtualized servers at low cost



Deploying in the Cloud

- Joyent is a cloud computing provider
- Virtualized servers at low cost
- Excellent performance



Deploying in the Cloud

- Joyent is a cloud computing provider
- Virtualized servers at low cost
- Excellent performance
- Adding resources as simple as filing a ticket!
 - Often doesn't require a reboot



Horizontal Scaling

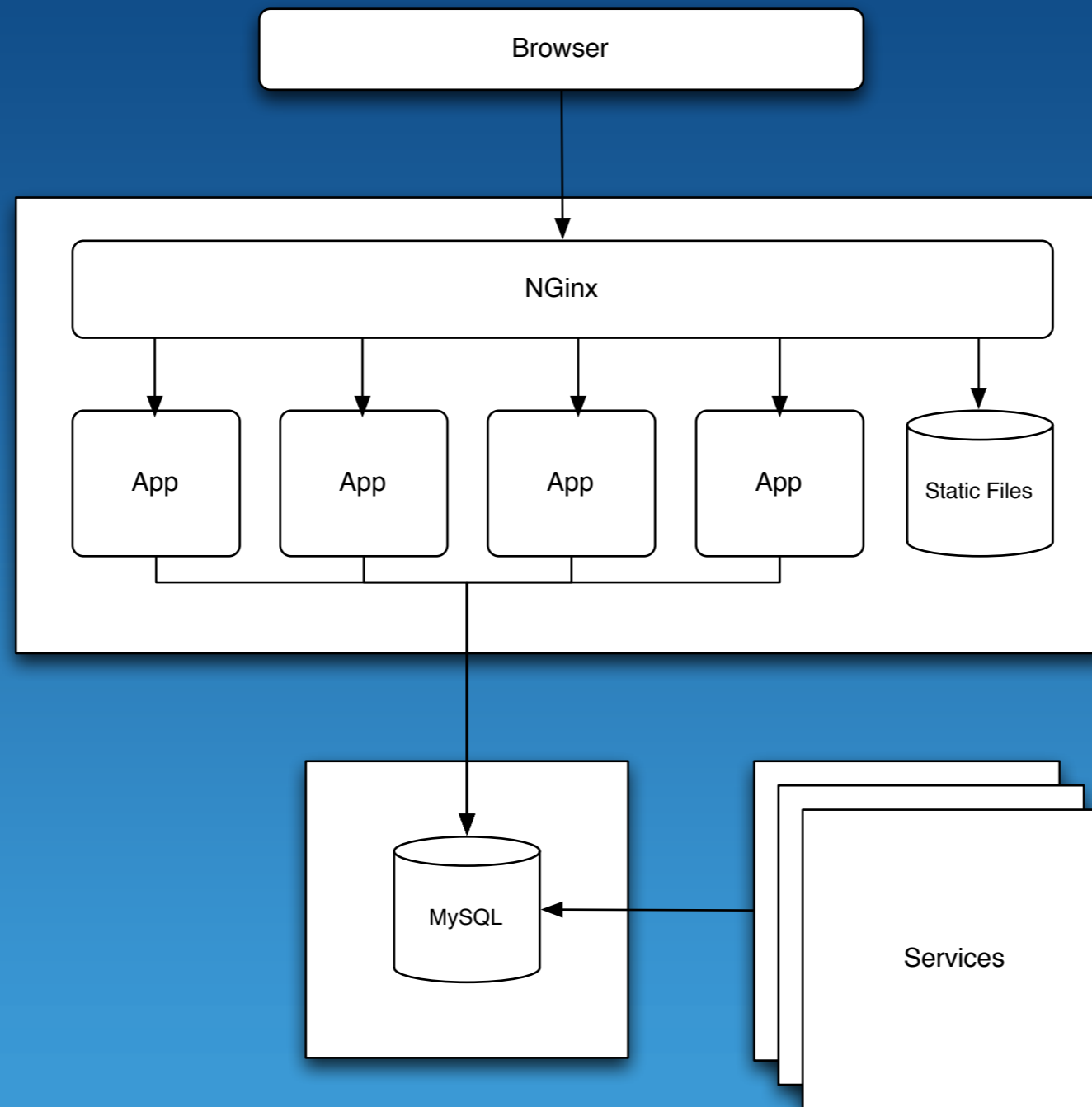
Problem: Separation of Concerns

Separation of Concerns

- Application, database, services, and website on same server
- Competition for resources

Separation of Concerns

- Application, database, services, and website on same server
- Competition for resources
- Virtualized servers are cheap – split them out!
 - Application “node”
 - Database
 - Services – email, billing, other daemons.



Improved Infrastructure

Problem: Scaling App Horizontally

Scaling App Horizontally

- Multiple servers presents several problems:

Scaling App Horizontally

- Multiple servers presents several problems:
 - How do we divide up requests?

Scaling App Horizontally

- Multiple servers presents several problems:
 - How do we divide up requests?
 - How to handle application session state?

Scaling App Horizontally

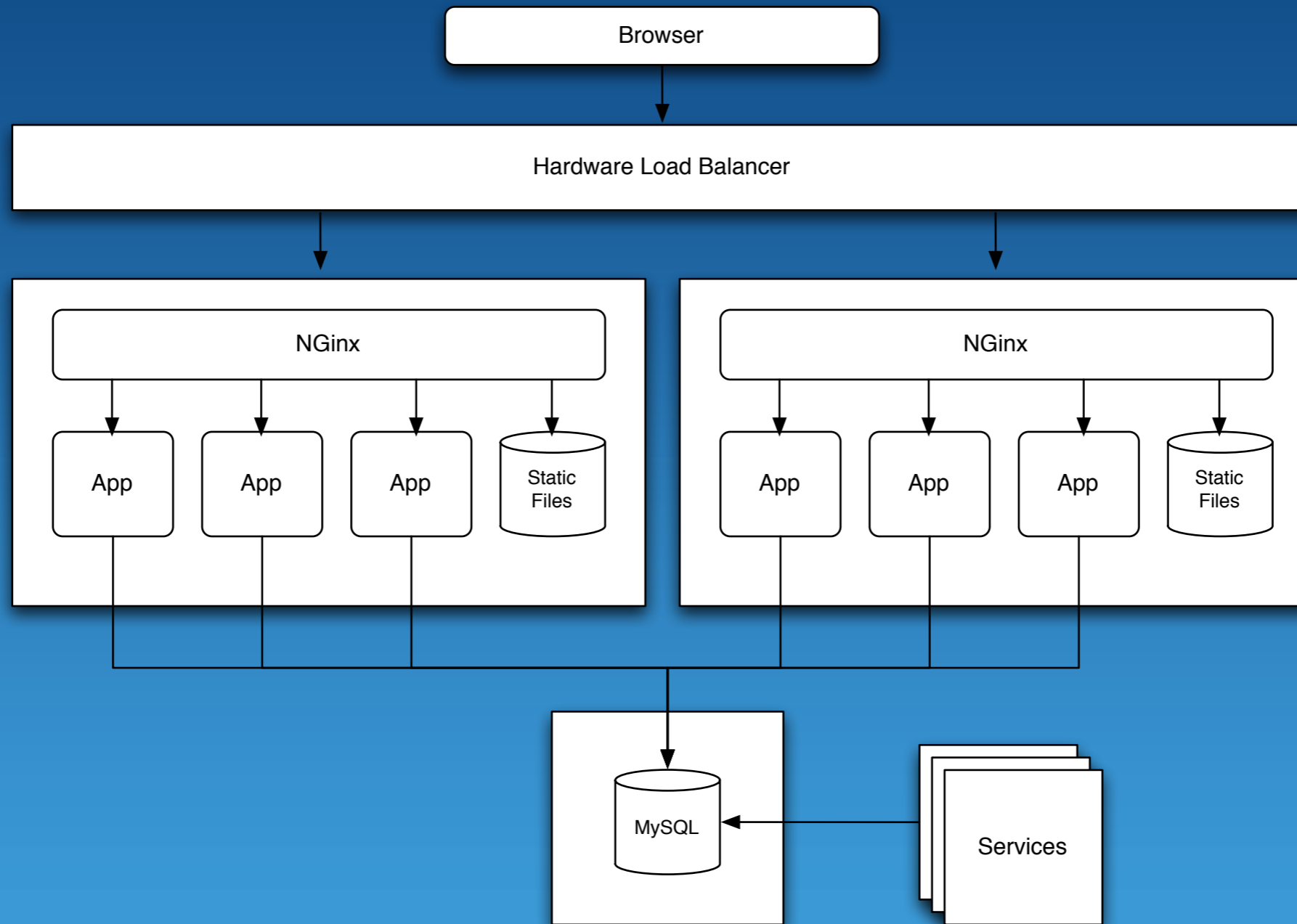
- Multiple servers presents several problems:
 - How do we divide up requests?
 - How to handle application session state?
 - Often stored in-memory or on disk.

Hardware Load Balancer

- Joyent hardware load balancer (BigIP)
 - Similar to Nginx, but in hardware
 - Load balances requests to a cluster of “nodes”
- New nodes can be added on-demand
- Added benefit: SSL acceleration in hardware

Cookie-Backed Sessions

- TurboGears sessions handled by “beaker”
- Beaker supports cookie-backed sessions
 - Encrypted, signed, secure
 - State lives in browser
 - Allows application to be “stateless”



Redundant Infrastructure

Benefits of Multiple Nodes

- Application redundancy
 - One node fails, the second automatically handles requests

Benefits of Multiple Nodes

- Application redundancy
 - One node fails, the second automatically handles requests
- Deployment causes less downtime
 - Rolling updates can be applied
 - Minimal disruption for users

Problem: Scaling the Database

Scaling the Database

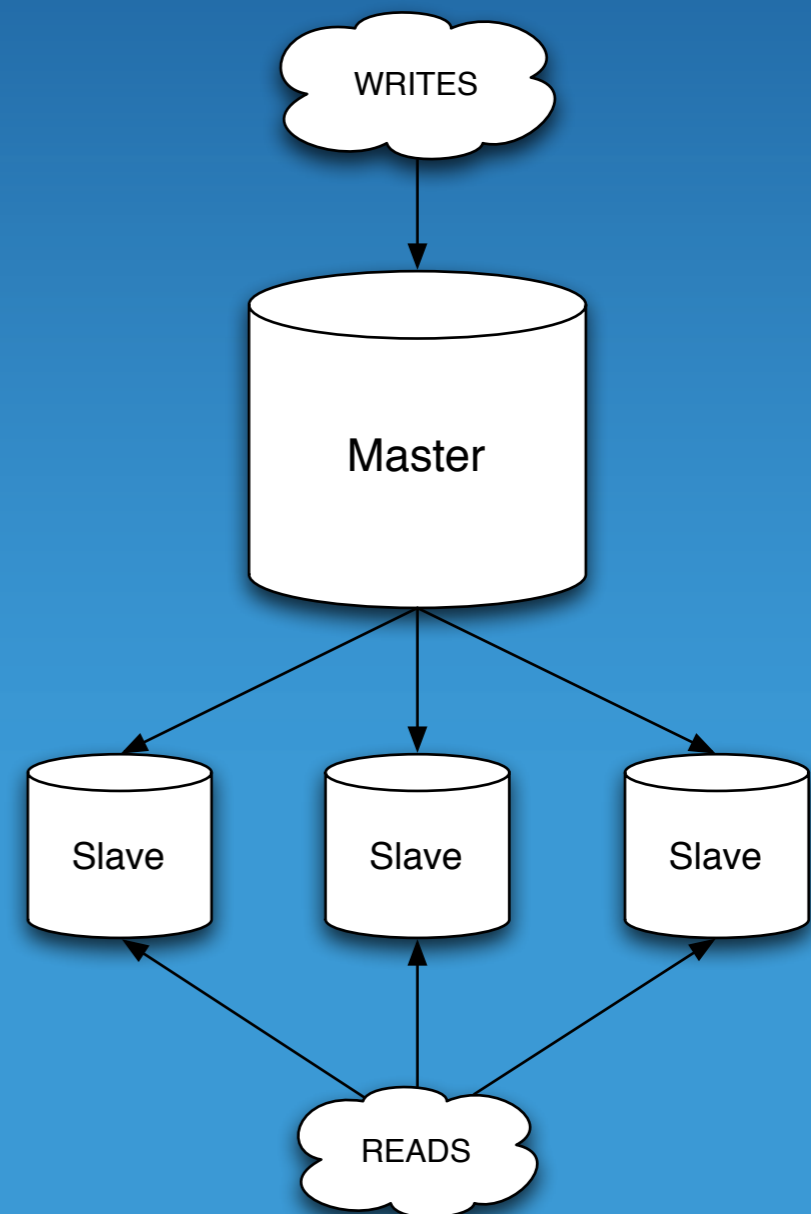
- ShootQ is heavily read-based
- Most requests do not modify data
- Database can be scaled vertically by adding resources
- What happens when we run out of resources?

MySQL Replication

- MySQL provides “master-slave” replication
- Multiple instances of your database
 - Master – read/write
 - Slaves – read-only copies

MySQL Replication

- MySQL provides “master-slave” replication
- Multiple instances of your database
 - Master – read/write
 - Slaves – read-only copies



Splitting Reads and Writes

- All SQL writes must be sent to the Master
 - How can we split reads and writes?

Splitting Reads and Writes

- All SQL writes must be sent to the Master
 - How can we split reads and writes?
- Piggyback on top of HTTP
 - GET and HEAD are defined as “idempotent”
 - POST, PUT, and DELETE can have side-effects

Splitting Reads and Writes

- ShootQ Database WSGI “middleware”
 - GET and HEAD are sent to the MySQL Slave

Splitting Reads and Writes

- ShootQ Database WSGI “middleware”
 - GET and HEAD are sent to the MySQL Slave
 - POST, PUT, and DELETE are sent to the MySQL Master

Splitting Reads and Writes

- ShootQ Database WSGI “middleware”
 - GET and HEAD are sent to the MySQL Slave
 - POST, PUT, and DELETE are sent to the MySQL Master
 - Requests to the master are also wrapped in a transaction

Splitting Reads and Writes

- ShootQ Database WSGI “middleware”
 - GET and HEAD are sent to the MySQL Slave
 - POST, PUT, and DELETE are sent to the MySQL Master
 - Requests to the master are also wrapped in a transaction
 - Errors cause transaction to be rolled back automatically

Slave Lag

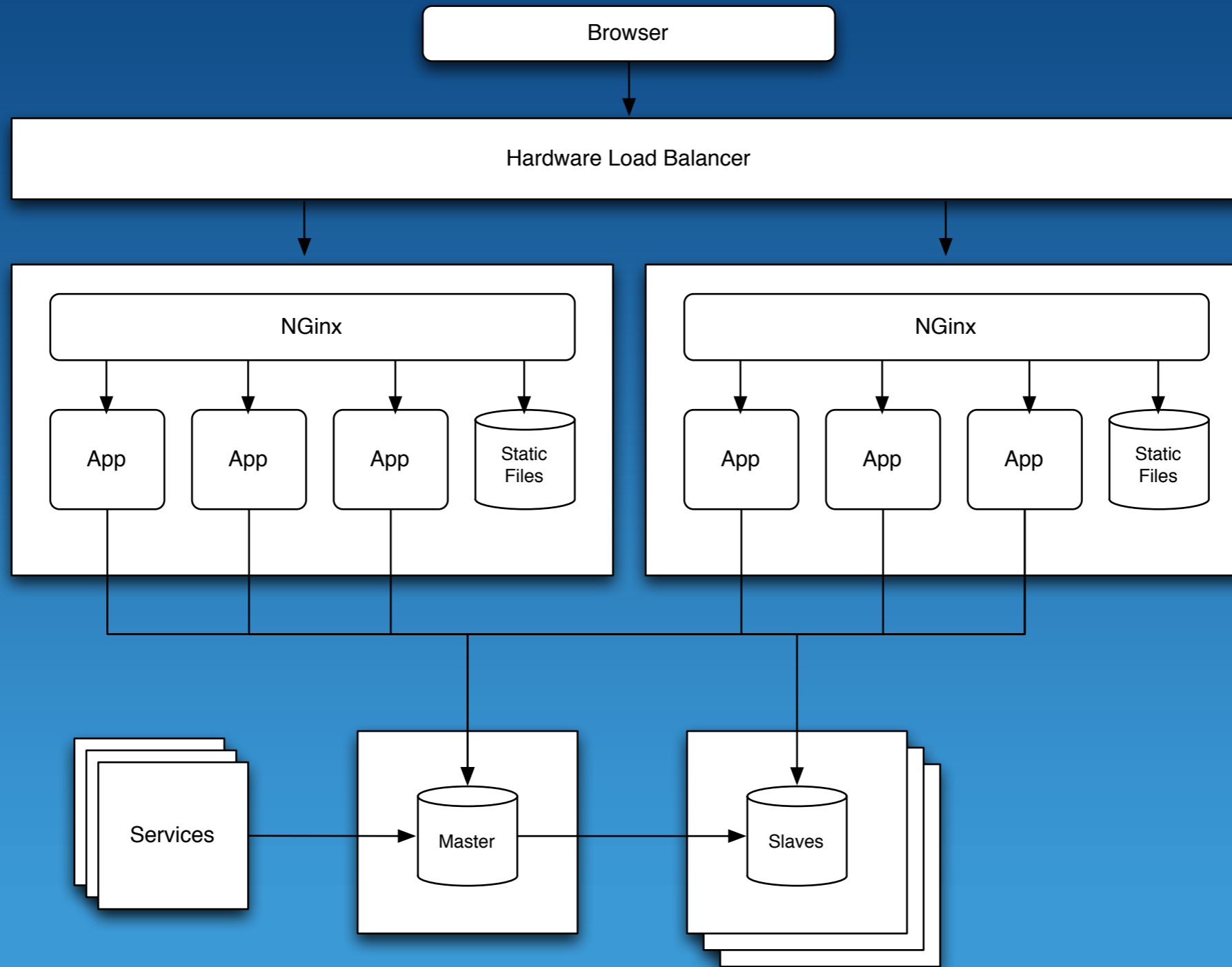
- Slave can “lag” behind Master
 - Writes followed by reads
- Solutions:
 - Decorator and Utility Function
 - WSGI middleware

Slave Lag

- Slave can “lag” behind Master
 - Writes followed by reads
- Solutions:
 - Decorator and Utility Function
 - WSGI middleware

```
@expose()
@transact_next(['/account'])
def update_account(self, name):
    account = get_account()
    account.name = name
    redirect('/account')

@expose()
def create_page(self, content):
    page = model.Page(content)
    uri = page.getURI()
    register_transact_next([uri])
    redirect(pageURI)
```



Final Infrastructure

Summary

Summary

- Select the right tools
 - Comfort
 - Applicability to task

Summary

- Select the right tools
 - Comfort
 - Applicability to task
- Scale vertically
 - Multi-process your application
 - Serve static content from a web or proxy server
 - Force static content to cache

Summary

- Select the right tools
 - Comfort
 - Applicability to task
- Scale vertically
 - Multi-process your application
 - Serve static content from a web or proxy server
 - Force static content to cache

Summary

- Select the right tools
 - Comfort
 - Applicability to task
- Scale vertically
 - Multi-process your application
 - Serve static content from a web or proxy server
 - Force static content to cache
- Scale horizontally
 - Separate concerns: database, application, services
 - Push session state to client-side
 - Employ a hardware or software load balancer to support multiple application servers
 - Make use of database replication

Summary

- Select the right tools
 - Comfort
 - Applicability to task
- Scale vertically
 - Multi-process your application
 - Serve static content from a web or proxy server
 - Force static content to cache
- Scale horizontally
 - Separate concerns: database, application, services
 - Push session state to client-side
 - Employ a hardware or software load balancer to support multiple application servers
 - Make use of database replication

Summary

- Select the right tools
 - Comfort
 - Applicability to task
- Scale vertically
 - Multi-process your application
 - Serve static content from a web or proxy server
 - Force static content to cache
- Scale horizontally
 - Separate concerns: database, application, services
 - Push session state to client-side
 - Employ a hardware or software load balancer to support multiple application servers
 - Make use of database replication
- Deploy into a cloud



What's New

- NEW** Live "Getting Started" webinar today Thursday 2/18 at 1pm Eastern 10am Pacific. You'll be able to sign in 5 min. before webinar! [Learn More](#)
- One or more email(s) failed to send.
- You have 8 new lead(s).
- You have 6 booking(s) to review.
- LaCour - Demo referred you to [Julie Simmons Wedding - Jun 20, 2009](#).
- FEB 17** LaCour rejected your referral for [Don Draper Dj Gig](#).
- FEB 09** Your questionnaire for Don Draper Wedding has received a response.
- FEB 04** Your most recent proposal for Bonnie Tyler Wedding was declined.
- JAN 25** Your questionnaire for Travis Schreer Wedding has received a response.

Notify All Users

What's Happening

Thu Today
Abernathy-Johnson - sales session

Fri February 19th
No scheduled events.

Sat February 20th
No scheduled events.

Sun February 21st
No scheduled events.

Mon February 22nd
Adam Linke Wedding - e sesh

Tue February 23rd
No scheduled events.

Wed February 24th
No scheduled events.

Scheduled Shoots

What's Next

Show: Next Task for: All Brands

Next Task	Due
Sugar - Client Approves Design <i>Leather Album -> Waiting on Client Approval</i>	96 days ago
Jose Rivera Wedding - Design Album <i>Cypress Fine Art Album -> Album Design</i>	96 days ago
Murphy-Smurf Wedding - Place order for Product	96 days ago

CLEVERtrain

pro training

LIVE+LEARN

cleverdevil.org/train