Tools of the Trade

Web Development at Fairfax Media

Who am I?

George Wright

Technology Director - Metro Publishing

Fairfax Media

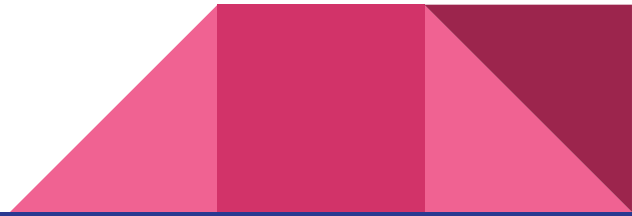
I am responsible for a delivery team that provides a core platform.

Janette Chiu

Software Engineer - Metro Publishing

Fairfax Media

I am responsible for software design and implementation.



I apologise in advance

- Software architecture is evolving rapidly and is extremely diverse.
- There are no right answers!
- Technology is very cyclical in nature
  - thin client -> fat client -> thin client
  - Centralised -> decentralised
- One area that seems to have more accepted is that open standards has won.

Software Developers

The CRAFT of software

- part engineering - accuracy and efficiency
- part designing - elegance
- part project managing - accountable
- part poetry - expressive

Microservices are the new norm.

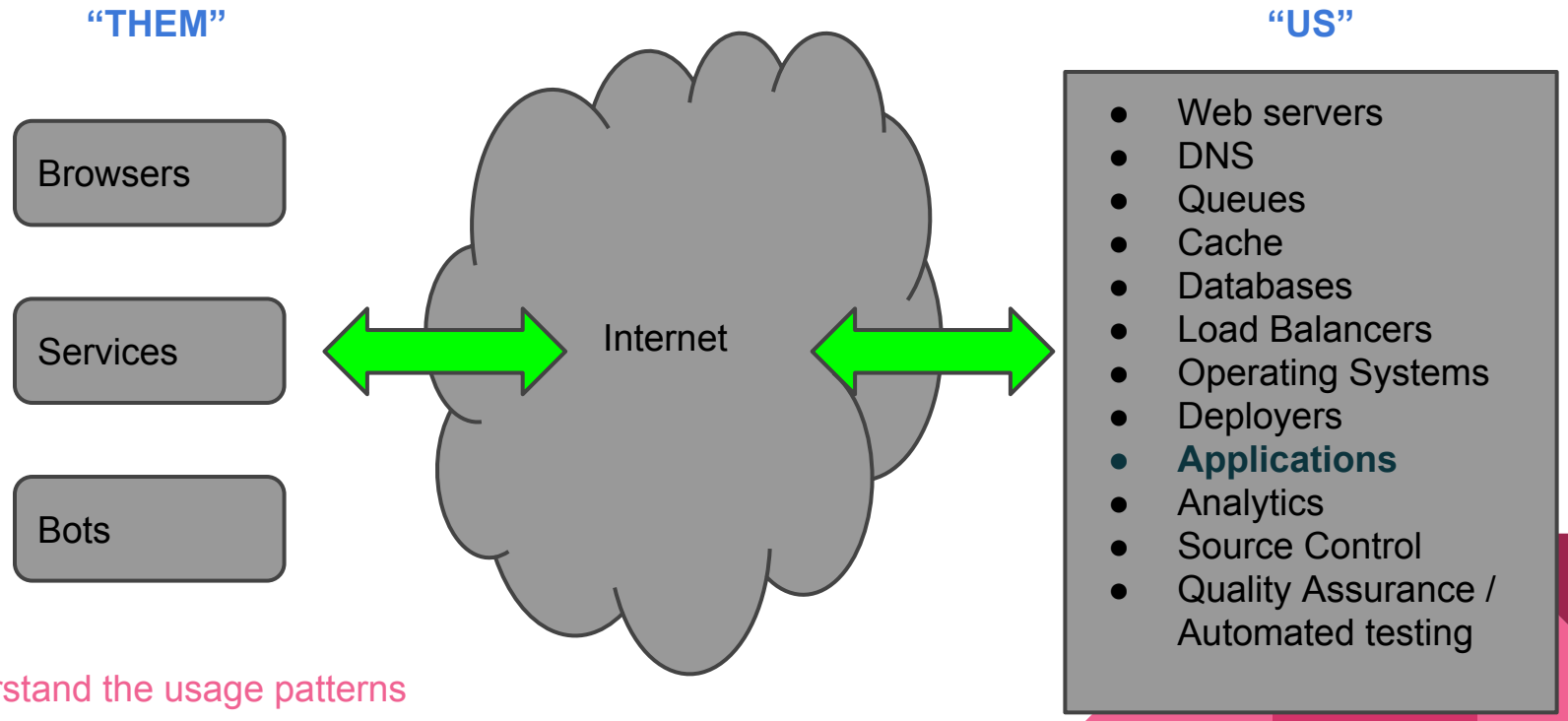
- Engineering for interoperability is key.
- No one technology is the silver bullet.
- Automated testing.
- Continuous Integration/Deployment.
- Serverless where possible. OUTDATED??
- No longer a monoculture approach. OUTDATED??
- Is event sourcing next?

the “Stack”

All of the systems and services that fulfil a request (end-to-end) - in this case a request from a web client.

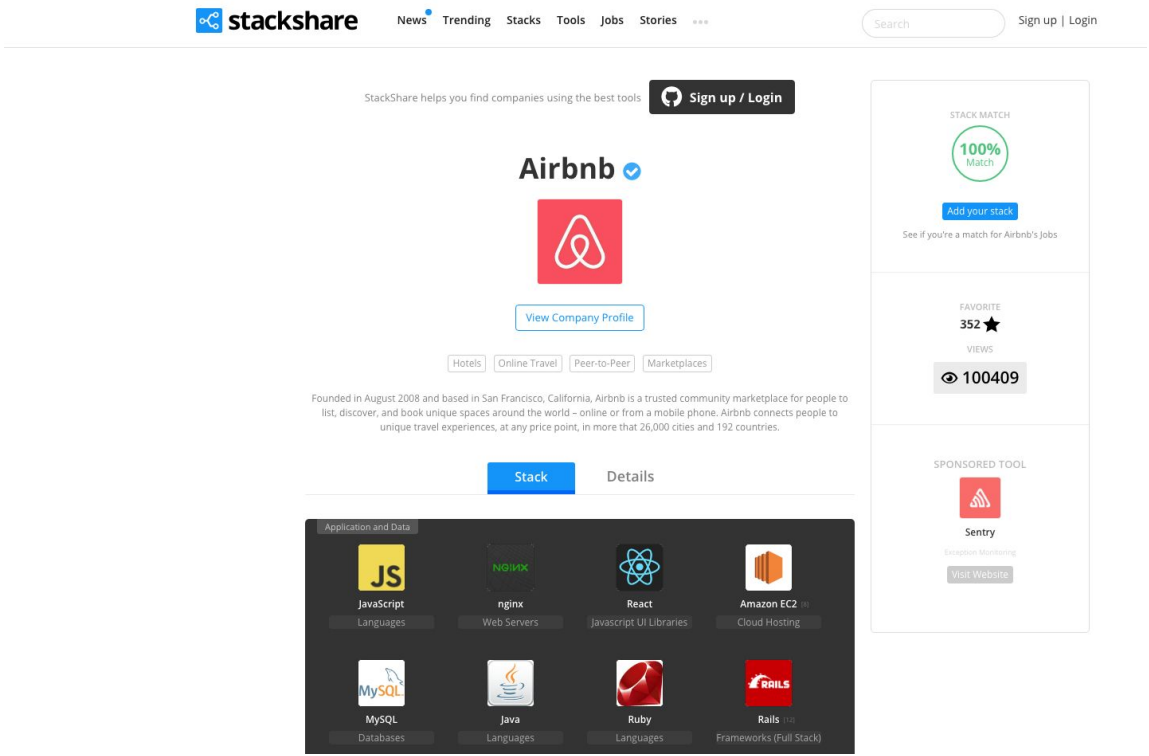
The stack is the largest box you can draw around the things you can control/influence.

Stack basics

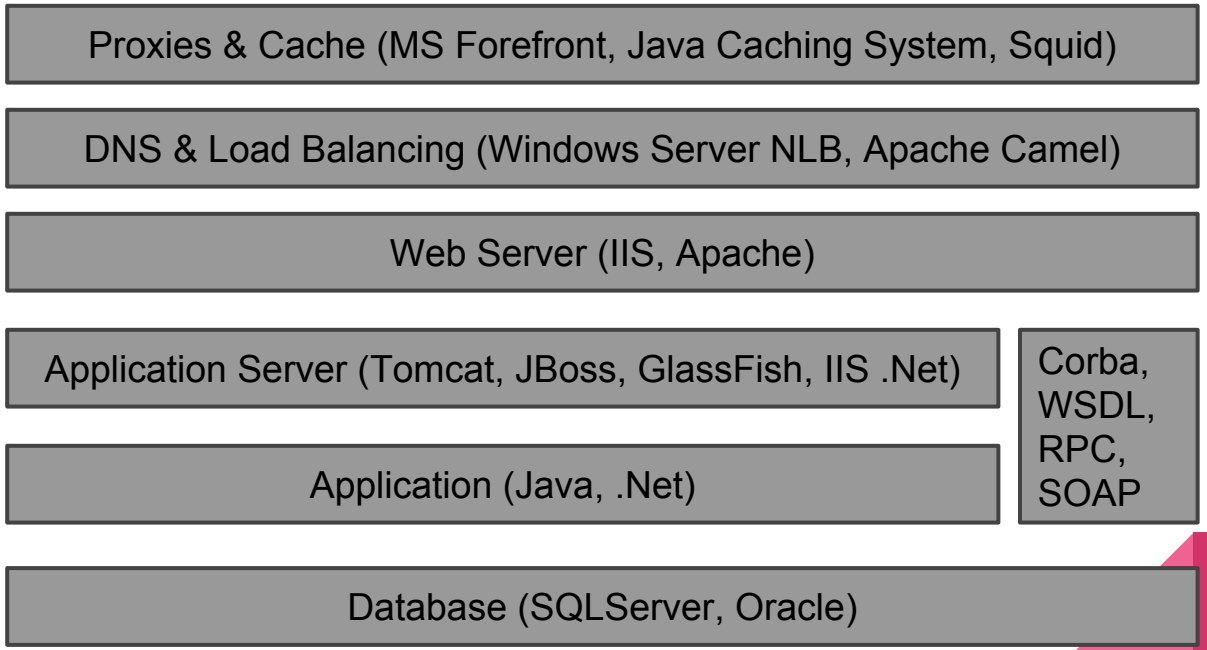


Understand the usage patterns

Stack Shopping



The “Corporate” stack



The “hipster” stack

MEAN stands for:

MongoDB is the leading NoSQL database, empowering businesses to be more agile and scalable.

Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications.

AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications.

Re-Platform

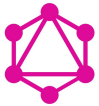
Language: GoLang

Release Pipeline: Concourse

Runtime environment: Docker/Kubernetes

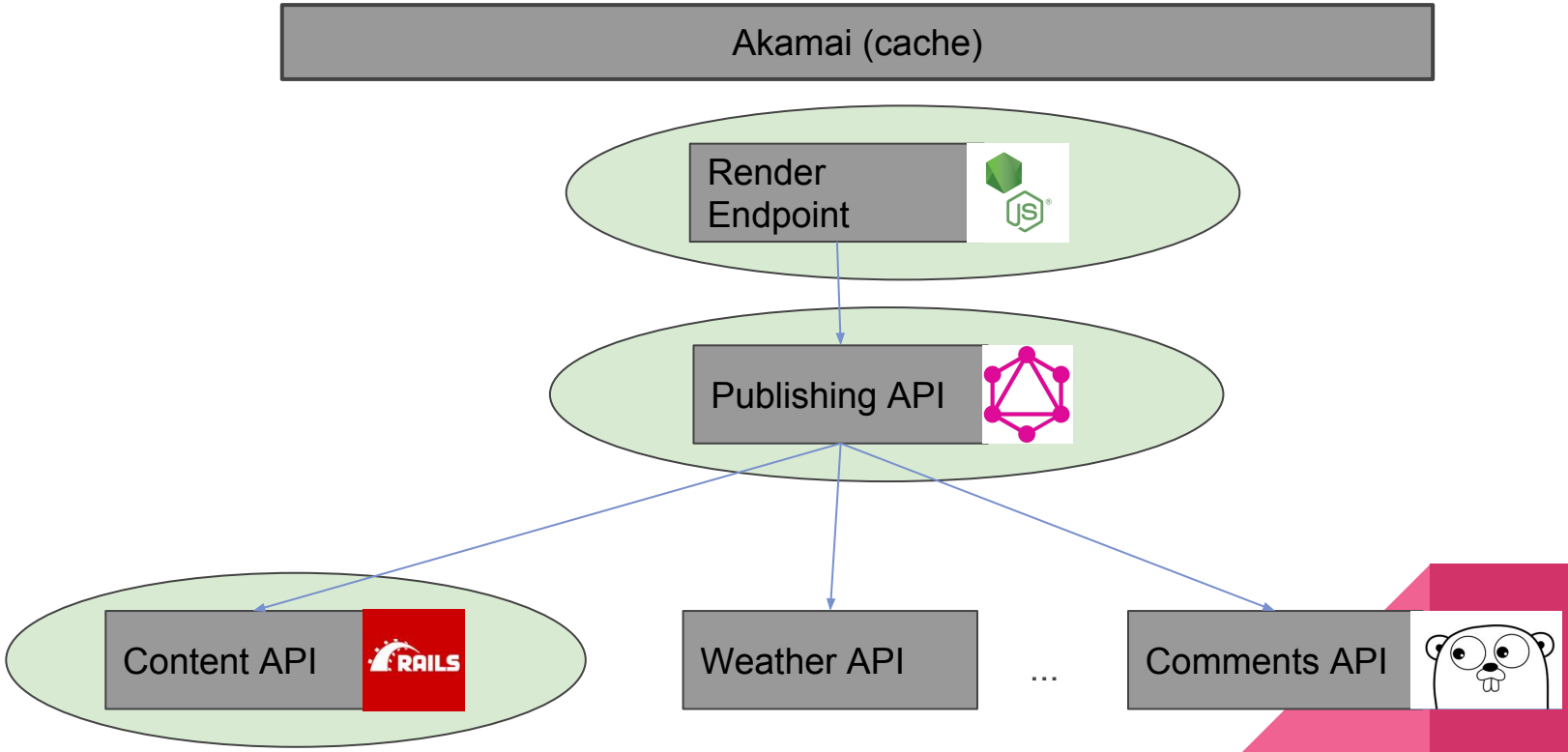
Message Streaming: AWS Kinesis

API Contract: GraphQL, gRPC



DOM rendering: React

Microservices



Amazon AWS

Rethinking the architecture at scale:

- Compute (EC2, Lambda) - processing
- Elastic Load Balancer
- Storage (S3/Glacier)
- Database (DynamoDB/RDS/Redshift)
- In Mem. Cache (Elasticache)
- DNS (Amazon Route53)
- Web Cache (CloudFront)
- Queues (SQS)
- Notification (SNS, SES)
- Streaming (Kinesis)
- Analytics (EMR)
- Applications (ElasticBeanStalk)

+ Many, many more.

Alternatives

Application Hosting - Heroku, EngineYard, OpenShift, Google Cloud Platform, Microsoft Azure, Joyent + new entrants every day.

Check out DigitalOcean tutorials

https://www.digitalocean.com/community/tutorial_series

The Browser “Basics”

- HTML5
- CSS3
- Javascript - JQuery(?), Modernizr
- Debugging - FireBug, Chrome Dev Tools
- JSON - CORS and JSONP
- REST

Frontend Stack provides:

- Package Management
- Templating
- Testing framework
- Common widgets
- Databinding
- Server communication

But wait there’s more..

Frontend JS frameworks (just a sample)

Yeoman: **Package manager:** yeoman, **Module system**: RequireJS, AMD,**Build system**: Grunt,**Widgets**: Bootstrap

Angular/Closure: Module system: Closure Library modules, Build system: ClosureBuilder, Templates: AngularJS, Data binding: AngularJS, Sync: AngularJS services, Widgets: Closure Library, Test framework: Closure Library testing

Foundation: Package manager: Component, Module system: CommonJS, Build system: Component, Data binding: Reactive or Rivets or Knockout or Backbone, Sync: component/model, Test framework: test/assert, Mocha

React, Vue.js etc.

Developer stack

Every developer should have their own personal productivity stack including:

- Editor - Notepad++, Atom, Sublime, vim etc.
- IDE - Visual Studio, JetBrain, Netbeans, Eclipse
- Version Control - git, mercurial, bzr, cvs, svn
- Repository - gitorious, github, bitbucket, launchpad
- Issue tracking - JIRA, Redmine, Bugzilla
- Agile/SCRUM - Greenhopper, Trello

Quality assurance stack

- Automated testing eg. Mocha, Jasmine, Selenium etc.
- User acceptance testing tools
- Production monitoring
- Incident management
- Disaster recovery plan
- Peer reviewed documentation
- Stakeholder management
- Legal sign-off(?)

Advice for new developers

- Code is disposable but features are not!
- “Perfect” code. Substance vs Purpose and excellentia (scale) vs perfectio (absolute).
- Practice Fermi Estimation techniques.
- Understand the difference between classical **C**riticism and **c**riticism.
- Right tool for the right job.
- Participate.
- Stay up to date with the trends.

Thank you UNSW

Questions?