Software is Eating the World, APIs are Eating Software

Steven Willmott
3scale Inc.
@njyx, @3scale
3scale is...

API Infrastructure Provider
Power 350+ APIs
110,000 Developers writing Apps

API Tech Operations  API Business Operations  Developer Support

3scale.net
What do you mean API?
Software is Eating the World

@pmarca - WSJ / Aug 2011
Almost every major industry is becoming software driven

Retail  Video  Telephony  Music

...
Amazon: “Software Driven Retail”

Lytro: “Software Defined Cameras”

Pixar: “Software powered animation”

Philips Hue: “Software Controlled Lighting”
(Web) APIs Are Eating Software

(The Web is Eating Software)
1. APIs are the key glue that make this software *remotely addressable*

2. APIs provide a myriad of new external building blocks to *speed up* and *enrich* software development

3. If you do these things together *special things happen*
Inside-Out
APIs Make Software Remotely Addressable
**London Transport**: Open Data

**Netflix**: “Massive Distribution”

**Evernote**: “Platformization”

**Johnson Controls**: “Software Controlled Buildings”
Outside-In
APIs are the new Libs

“In computer science, a library is a collection of implementations of behavior, written in terms of a language, that has a well-defined interface by which the behavior is invoked”
Download & Add to Classpath

Became

Find and Integrate
Speed (Time to Market)

*e.g.*

![shirts.io](image1)

+ Devops Borat

*e.g.*

![liveplasma](image2)

(Top Mashup on Progr Web)
Richness / Functionality

e.g. Stripe

AccuWeather.com
Offboarding

e.g. Effects SDK (Rich Photo Effects For Mobile by SDK)

E.g. New Relic (Monitoring System in the Cloud)
Reliability

e.g. SendGrid (Email by API)

e.g. S3 (Amazon S3)
Mixing the Two
Software Development over Time

Software Before 1995

Software 1995 – 2010

Software 2010 -
Enables specialization, focus, much wider distribution
1. Changing Software Development
Software as Model-View-Controller

MODEL = DATA

VIEW = FORM

CONTROLLER = BUSINESS LOGIC
Now it can done at the company / organization level
Example “Models”

Model

Data Anywhere in any form

The New York Times

xignite

Bloomberg

data.gov.uk

WeatherBug
Example: Views

View

Many Delivery Channels

Flipboard

Yahoo!
Example: Controllers

Controller

Many Delivery Channels
APIs Enable Separation & Focus

Model
Data Anywhere in any form

View
Many Delivery Channels

Controller
Third parties operating on data

Distributed Applications
Software + APIs allow Businesses to co-evolve much faster
Fundamentally Different Model of Developing Software

(MVC is only one model – the point is: componentization is possible)
II But this is Hard to Do
Downsides to Opening APIs...

- Security is key
- Scalability needs to be built in
- It requires long term support
- New type of business interaction

- Vendors can Help
- In many cases some of the work is already done
Downsides to using external APIs...

- Latency?
- Availability?
- Security?
- SLAs?
- Cost
- Service Continuity?

- In most cases there are no other ways to solve the problem
- Tools are emerging
This is Distributed Systems Engineering
10 Hard Things About Building Distributed Systems

- Interface Definition & Consistency
- Latency
- Slow or Dead?
- Distributed State Synchronization
- Remote Clock Problems
- Error Detection
- Change Management
- Static & Dynamic Testing
- Code Validation / Verification
- Frame Problems
But Wait...

It’s even harder than that ...
Distributed Across Organizational Boundaries

- No access to source code
- No knowledge of Server Environment
- Security and Access Permissions Everywhere
- Identity Problems
- Shared Semantics are much harder to achieve
- Unknown / Mismatched Scale Issues
- Danger of Much Wider Interdependencies – Frame problem is worse!

3SCALE
Lots of Challenges
What does this mean for The Web and Web Information Systems?
When Building Something - what do you care about?
A: My System
A: My System
A: My Ecosystem

(Everything which is integrated with me)
## Getting the Question Right

<table>
<thead>
<tr>
<th>My System</th>
<th>My Ecosystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Architecture</td>
<td>• Who is integrated</td>
</tr>
<tr>
<td>• Security</td>
<td>• How do they do this</td>
</tr>
<tr>
<td>• Scaling</td>
<td>• Management of Dependencies</td>
</tr>
<tr>
<td>• Data Models</td>
<td>• Standard Interfaces</td>
</tr>
<tr>
<td>• Interactions</td>
<td>• Think about Client needs to rethink transaction loads</td>
</tr>
<tr>
<td>• Consistency</td>
<td>• Share scaling burden with developers</td>
</tr>
</tbody>
</table>

- More Obvious
- Less Obvious
We’re Distributed

- APIs/REST/ SOAP
- Services Model
- Ecosystems
- Dependencies
Problems

1. Mixed Formats
2. Poor Designs
3. No Discovery
4. Fragile Integrations
5. Code Overhead

A Lot of Work to do!
Conclusions

Your Web Systems are now Distributed Computing

The World is being Software Enabled and Web-API-ified

Long way to go - but the basics are having a BIG impact
Discussion Topics

- Hypermedia
- API Growth
- APIs & HTML5
- SOAP v's REST
- APIs v's Websites
- APIs & Semantic Web
- ...

3SCALE
3scale: http://www.3scale.net
APITools: http://www.apitools.com
APICodex: http://apicodex.3scale.net
Thank You!

Contact:
http://www.3scale.net
@njyx - steve@3scale.net
APIs enable componentization across organizational boundaries