Cloud First Application Development

Ivy+ Human Resources Information Systems

October 27, 2017

Bryan Hopkins
Who is this guy?

Bryan Hopkins – University of Pennsylvania

- App Dev Lead for Penn’s Cloud First team
  - Past Lives:
    - Cold Fusion developer (ewww)
    - Java developer
    - Architect on our frameworks team
    - Infrastructure team lead for our initial Banner implementation
    - Technology Coordination lead for campus initiatives
  - Keep it simple, with the support team in mind
  - “The State Department within the SWAT Team”
Core Cloud First Team

Not everyone is speaking, but this is everyone’s work

- Tim Bouffard, Application Architect
- Sam Donnelly, Sr. Application Developer
- Justin Ettore, Systems Architect
- Dane Fetterman, IT Architect
- Tiffany Hanulec, IT Technical Director
- Janet Lind, IT Director
- Anome Mammes, Sr. Application Developer
- Lisa McBriar, Sr. Business Systems Analyst
- Matt Schleindl, Application Architect
- Namitha Venkatesh, Senior IT Manager
- Mathias Wegner, Network Engineer
Today

- What is an application?
- Where did Penn ISC start, and why change?
- What is Cloud First?
- How are we working?
- What have we learned?
What goes into an app

- A user interface
  What people see
But there’s a lot more lurking…

...Technical Debt
What goes into an app

- Business logic
- Actual value
What goes into an app

✓ AuthN and AuthZ
Security
What goes into an app

- Persistence and data stores
  Making it all matter
What goes into an app

- Integration points
  Playing well with others
What goes into an app

✓ Testing
   Proof it’s doing what it should be
What goes into an app

- A source repository
- Storage for code and config
What goes into an app

✓ Deployment pipeline
  Getting it built and available
What goes into an app

✓ A platform
  Somewhere to run and scale
That’s a lot of stuff

✓ User interface
✓ Business logic
✓ Authentication and security
✓ Persistence and data stores
✓ Integration points
✓ Testing
✓ Source repository
✓ Deployment pipeline
✓ Platform and Infrastructure
✓ Process and SDLC
By the numbers

- 1000+ Applications
- 90+ Customer organizations
- 300+ skills and tools
- Less than 40 developers and architects
- Only 3 framework contributors
- 5 Database Administrators
- 9 Systems Administrators

To support and run it all!
➤ You can be on the hook for many of these, even if it’s a vendor product!

➤ Particularly if you’re hosting the application
  □ Security
  □ Testing
  □ Platform and Infrastructure
  □ Repository
  □ Deployment
And...
We are being judged...

- Increasingly for our...
  - Speed!
  - Accessibility (usability for those with disabilities)
  - Responsiveness (usability on a wide range of devices)
  - Cost transparency

- Less for our...
  - Risk profile
  - Meticulous custom services

👉 Less is not none!
So what is Cloud First?
It’s a gift: Green field
In short:

- Make it easier to choose Software as a Service options
- Make it easier for partners to work with us
- Re-envision the full technology stack for cloud-native agility
- Model and socialize iterative and transparent value delivery
- Automate or eliminate maintenance work
- Socialize and share what we learn
Principles

- Choose simplicity, speed, and iterative improvement over up-front completeness and strict controls.

- Continuously iterate and re-evaluate decisions; nothing is beyond question.

- Be mindful of increasing technical debt.

- Choose automated solutions over manual ones, but respect the implications to manual process owners.

- Use tools and technologies “off the shelf” in secure ways and as the provider intended.
So how does that translate?

✓ User interface
✓ Business logic
✓ Authentication and security
✓ Persistence and data stores
✓ Integration points
✓ Testing
✓ Source repository
✓ Deployment pipeline
✓ Platform and Infrastructure
✓ Process and SDLC
So how does that translate?

- Assembling a stack
- Prioritize
- Evaluate
- Integrate
- Automate
ISC’s Cloud First Program at a glance

James Choate
Owner

Tiffany Hanulec
Architecture, Infrastructure & Migration

Bryan Hopkins
Application Design and Tools

Teresa Leo
Communications and Engagement

Janet Lind
Integration

Core Team
Application Design & Tools
Architecture, Infrastructure, & Migration
Integration
Communications & Engagement

Org Transformation
Contracts & Procurement
Security
Funding & Finance
IAM
Client Engagement
For the AppDev Team...

- We built these ourselves
  - Full control
  - Minimal short-term risk
  - Learn one language
  - Unique ISC terminology
For the AppDev Team...

- Someone else built it
  - Less control
  - More short-term risk
  - Learn to learn technologies quickly
  - Common terminology
  - More time spent on what people see, not frameworks and platforms!

- Assembling a cloud native application delivery stack
- Pioneering service-oriented, agile processes with automation
- Pioneering mobile-first and accessible designs
For the Platforms Team...

- Compute resources are
  - Carefully maintained
  - Built individually from images
  - Managed by hand
    - According to policy
  - In Penn’s data center
  - Scaled to meet peak usage
For the Platforms Team...

- Compute resources are
  - Disposable and read-only
  - Idempotently deployed
  - Managed as code
    - According to policy
  - In the public cloud
  - Scaled to meet demand

- Migrate existing virtual workloads to cloud IaaS providers
- Enable fast and secure connectivity with cloud providers
- Enable cloud-native application delivery stack
For the Integrations Team...

- Integrations
  - Transform and migrate data
  - Point-to-point as needed
  - In many tools and languages
  - Coded by developers
  - Focused on Oracle databases
  - Prefer flat files and direct database links
For the Integrations Team...

- Integrations
  - Transform and migrate data
  - Coherently structured
  - Managed by a single tool
  - Configured by data analysts
  - Supports a wide range of data sources
  - Prefer RESTful web services and APIs with partners

- Implement cloud-based and cloud-enabled ETL solution
- Establish data governance and integration preferred practices for the cloud
Bottom Line For All Teams…

Make us nimble integrators, and maintain low technical debt.
Our Process

1. **Blank Slate**
2. Select a few areas of focus
3. Update architecture and processes
4. Choose disposable placeholders
5. Retrospective
6. Evaluate / Pilot
7. Release to Production
This is a continuous process... but where are we now?
Architectural Roadmapping: More important than ever
**User Interface**

<table>
<thead>
<tr>
<th>For Developers…</th>
<th>For Analysts…</th>
<th>For Project Managers…</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Full framework</td>
<td>• Responsive by default</td>
<td>• Open source</td>
</tr>
<tr>
<td>• Thousands of components</td>
<td>• Accessible components</td>
<td>• From Google and Twitter</td>
</tr>
<tr>
<td>• Readily available examples</td>
<td>• Cleaner, simple UIs</td>
<td>• Easy to find resources</td>
</tr>
<tr>
<td>• Data binding</td>
<td>• UX vision more achievable</td>
<td></td>
</tr>
<tr>
<td>• In-browser debugging</td>
<td>• Inexpensive</td>
<td></td>
</tr>
<tr>
<td>• Angular-aware code completion</td>
<td>• Contractors already know it</td>
<td></td>
</tr>
<tr>
<td>• Variable completion with typing with JSDoc</td>
<td>• Inexpensive</td>
<td></td>
</tr>
</tbody>
</table>

![AngularJS](https://via.placeholder.com/150) ![Bootstrap](https://via.placeholder.com/150) ![WebStorm](https://via.placeholder.com/150)
# Business Logic

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Full framework</td>
<td>• <strong>Self documenting APIs</strong></td>
<td>• Open source</td>
</tr>
<tr>
<td>• Thousands of packages</td>
<td>• Automatically generated test interfaces</td>
<td>• Tremendous campus presence</td>
</tr>
<tr>
<td>• Readily available examples</td>
<td>•</td>
<td>• Very easy to learn</td>
</tr>
<tr>
<td>• Simple and powerful: few lines of code</td>
<td>•</td>
<td>• Easy to find resources</td>
</tr>
<tr>
<td>• Django-aware code completion</td>
<td>•</td>
<td>• Inexpensive with a free tier</td>
</tr>
<tr>
<td>• Great Django Testing integration</td>
<td>•</td>
<td></td>
</tr>
</tbody>
</table>

![Django and Python logos](images/django_python.png)

![JetBrains PyCharm](images/pycharm.png)
# Authentication and Security

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="nodejs.png" alt="Node.js" /></td>
<td><img src="npm.png" alt="npm" /></td>
<td><img src="webstorm.png" alt="Webstorm" /></td>
</tr>
<tr>
<td>- Thousands of packages</td>
<td>- No effort PennKey login with Shibboleth</td>
<td>- Open source</td>
</tr>
<tr>
<td>- Readily available examples</td>
<td></td>
<td>- Easy to find resources</td>
</tr>
<tr>
<td>- Incredible I/O performance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Full stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Framework-aware code complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Fullstack debugging with AngularJS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Inexpensive with a free tier</td>
</tr>
</tbody>
</table>
# Data Store

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Oracle-like features</td>
<td>• Drop-in replacement for Oracle</td>
<td>• Open source, ethical, and Free</td>
</tr>
<tr>
<td>• That actually work!</td>
<td>• Works great with Talend</td>
<td>• Strong local presence</td>
</tr>
<tr>
<td>• Highly performant</td>
<td>• DB of choice for warehousing</td>
<td></td>
</tr>
<tr>
<td>• Relational AND NoSQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Works with every RDBMS</td>
<td>• Very easy to use</td>
<td>• Inexpensive</td>
</tr>
<tr>
<td>• Powerful autocompletion</td>
<td>• Don’t need to know SQL</td>
<td></td>
</tr>
</tbody>
</table>

- PostgreSQL
- JetBrains DataGrip
## Integration

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fully supported SDK in every language</td>
<td>• Powerful and flexible security models</td>
<td>• Incredibly cheap</td>
</tr>
<tr>
<td>• Easy to follow examples</td>
<td>• API Keys and Usage plans</td>
<td>• Fully managed</td>
</tr>
<tr>
<td>• Framework and platform agnostic</td>
<td>• Configurable “stages”</td>
<td>• Serverless</td>
</tr>
<tr>
<td>• Built in DDoS protection</td>
<td></td>
<td>• Taggable for cost transparency</td>
</tr>
</tbody>
</table>

- Inexpensive
- Taggable for cost transparency

**API Gateway**

**Amazon SQS**

University of Pennsylvania ISC: Cloud First
## Integration

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Integrates directly with AWS Services like S3 and RDS</td>
<td>• No code to write or support</td>
<td>• Managed Software as a Service</td>
</tr>
<tr>
<td>• Fully automated</td>
<td>• Near real-time replication (5 minutes)</td>
<td>• Great support and community</td>
</tr>
</tbody>
</table>

[talend logo]
Closer Look: Microservices and SOA

- No longer self-contained
- Think service areas
  - DAR
  - Finance
  - Student systems
  - HR
  - Research
- Composed of small pieces
  - Reusable
  - Built that way
  - Deployed that way

Looser Coupling, More Flexible/Portable, More Complex Outer Architecture

Tighter Coupling, Less Flexible/Portable, Less Complex Outer Architecture
Closer Look: Microservices and SOA

- No longer self-contained
- Think service areas
  - DAR
  - Finance
  - Student systems
  - HR
  - Research
- Composed of small pieces
  - Reusable
  - Built that way
  - Deployed that way
# Testing

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Confidence in the spec</td>
<td>• Easy to create and understand</td>
<td>• Bugs caught early take 10x less time</td>
</tr>
<tr>
<td>• Built for AngularJS</td>
<td>• Fully automated regression tests</td>
<td></td>
</tr>
<tr>
<td>• Easy stubbing and mocking with Sinon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Straightforward</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Like Junit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• <em>Easy</em> to test</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WebServices: no setup</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**For Developers...**

- Confidence in the spec
- Built for AngularJS
- Easy stubbing and mocking with Sinon
- Straightforward
- Like Junit
- *Easy* to test
- WebServices: no setup

---

**For Analysts...**

- Easy to create and understand
- Fully automated regression tests

---

**For Project Managers...**

- Bugs caught early take 10x less time
# Source Repository

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Offline work</td>
<td>• Create documentation developers actually use</td>
<td>• Open source</td>
</tr>
<tr>
<td>• Easy branches and merging</td>
<td>• Intuitive web interface to browse files</td>
<td>• Universal adoption (git)</td>
</tr>
<tr>
<td>• Webhooks for CI/CD integration</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Gitlab Cloud**
## Deployment Pipeline

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Automate with Python</td>
<td>• More frequent deployment</td>
<td></td>
</tr>
<tr>
<td>• Developers’ Swiss Army knife</td>
<td>• Automated tests run on each build</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Slack Integration</td>
<td></td>
</tr>
</tbody>
</table>

*Jenkins*
Platform

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Serverless</td>
<td>• Incredibly cheap</td>
<td></td>
</tr>
<tr>
<td>• <strong>Only worry about code</strong></td>
<td>• Nothing to patch</td>
<td></td>
</tr>
<tr>
<td>• Provisioned as code</td>
<td>• Massively scalable</td>
<td></td>
</tr>
<tr>
<td>• Isolated microservices</td>
<td>• Taggable for <strong>cost transparency</strong></td>
<td></td>
</tr>
<tr>
<td>• Easy DRF integration with Zappa</td>
<td>• Can be sized and scaled to order</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Taggable for <strong>cost transparency</strong></td>
</tr>
<tr>
<td>Amazon Lambda</td>
<td></td>
<td></td>
</tr>
<tr>
<td>docker</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Platform**

*Amazon Lambda*

- Can run any docker image
- Service isolation
- Blue-green deployments
Platform

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fully Managed Service</td>
<td>• Access like any other database</td>
<td>• Fully managed</td>
</tr>
<tr>
<td>• Provisioned as code</td>
<td></td>
<td>• Nothing to patch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Scalable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Taggable for cost transparency</td>
</tr>
</tbody>
</table>

Amazon RDS
Closer Look: Infrastructure as Code

- Everything about the app is just code
- Creating it defines and documents it
- No patching. Delete and re-create.
- No waiting!
  - No manual action!
  - Right the first time!
Closer Look: Infrastructure as Code

```
"MyDB" : {
    "Type" : "AWS::RDS::DBInstance",
    "Properties" : {
        "DBSecurityGroups" : [
            {"Ref" : "MyDbSecurityByEC2SecurityGroup"}, {"Ref" : "MyDbSecurityByCIDRIPGroup"} ],
        "AllocatedStorage" : "5",
        "DBInstanceClass" : "db.m1.small",
        "Engine" : "MySQL",
        "MasterUsername" : "MyName",
        "MasterUserPassword" : "MyPassword"
    },
    "DeletionPolicy" : "Snapshot"
}
```
If everything is code, process from that world matters more
Closer Look: Cloud First Agile

**Terminology**

- **Backlog**
  Sorted list of everything the clients want. Things at the top are better defined than things at the bottom.

- **Sprint**
  Period of time where the team focuses on implementing the clients' top priorities.

- **User story**
  The basic unit of work capturing a feature or requirement from the end user's perspective.

- **Epic**
  A collection of User Stories that together define a significant feature or business process.
Closer Look: Cloud First Agile

Create backlog
- Create VERY high level story list

Prioritize and Define
- Pick top 1-2 stories
- Feature files
- Screens

Build and Iterate
- Technical subtasks
- Playbook
- Unit tests
- Daily Standup

Test and Deploy
- E2E Tests
- Able to launch?
- MVP?
Process and SDLC

<table>
<thead>
<tr>
<th>For Developers...</th>
<th>For Analysts...</th>
<th>For Project Managers...</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Familiar tool</td>
<td>• Plugin store</td>
<td>• Transparency</td>
</tr>
<tr>
<td>• Easy to use</td>
<td>• Dependency tracking</td>
<td>• Reports and progress-at-a-glance</td>
</tr>
<tr>
<td>• Highly collaborative</td>
<td>• Drag and drop prioritization</td>
<td>• Collaborate externally</td>
</tr>
<tr>
<td></td>
<td>• Focus is on “Yes”</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Focus on only a few details at a time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Frequent feedback and adjustment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Rely on domain knowledge and organization</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• No document writing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Speed and time-to-value</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Responsive to changes in priorities</td>
<td></td>
</tr>
</tbody>
</table>
Nota Bene

- There is more work going on than what’s discussed here!
  - Exploration into Microsoft Azure and Google Cloud Platform
  - Migrating workloads from Penn Datacenters to Cloud IaaS providers
  - Advanced configuration management
  - Many Software as a Service implementations
    - Microsoft Office 365
    - Workday
    - Banner / Pennant
    - More!
In Summary
A shift in risk, not a magic bullet

AWS S3 Outage – 2/28/2017

• Bad day for the cloud
• Developers in the dark
• Apps couldn’t restart
• Images didn’t load

Data Center Outages – 6/21, 28/2017

• Good day for the cloud
• Developers unaffected (but it was dark)
• Apps ran and could be built (thanks to Shibboleth distribution)
• Campus integrations degraded gracefully

Credit: The Register [https://www.theregister.co.uk/2017/03/01/aws_s3_outage/](https://www.theregister.co.uk/2017/03/01/aws_s3_outage/)

Credit: Matt’s iPhone
What’s next? Questions...

- Lowering the barrier of entry for API Services
- Increasing the synergy of API Services
- Improving test automation
- Improving build automation
- Improving resiliency
- Improving agile processes
- Train and empower our staff
- Keeping technical debt short term: evolving

... and whatever’s on your mind. Thank you!
Additional Thanks and Credits

The Full Cloud First Team
James Choate
Dave Curran
Katie Faucett
Tiffany Hanulec
Peter Heinemann
Teresa Leo
John Mulhern III
Frank Nguyen
Vasu Raman
Michel van der List