



Cloud First Application Development

Ivy+ Human Resources Information Systems

October 27, 2017

Bryan Hopkins

ISC's Cloud First Program



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?

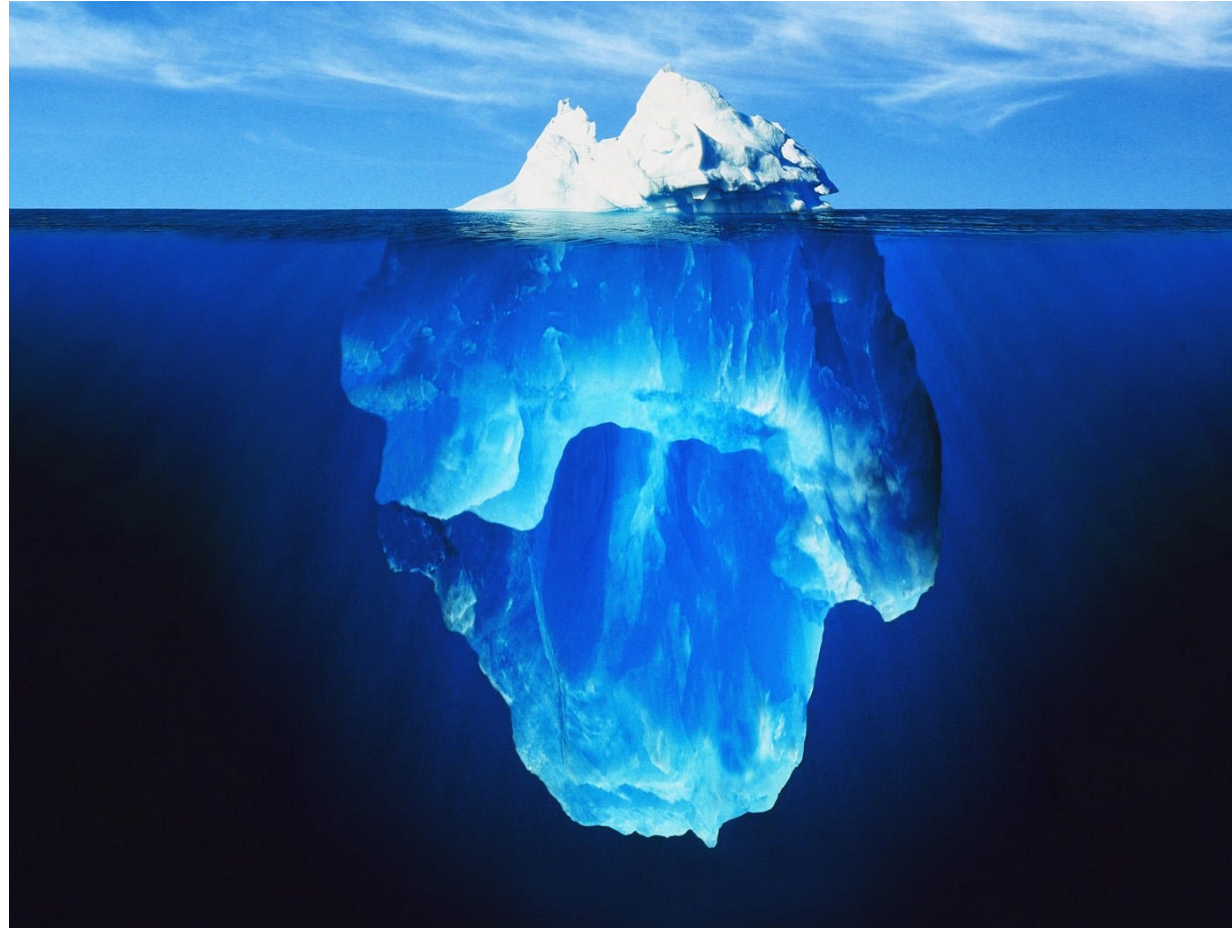


What goes into an app

- ✓ 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



The image shows a screenshot of the Penn WebLogin page. At the top, there is a blue header with the University of Pennsylvania logo and the text "Penn UNIVERSITY of PENNSYLVANIA". Below the header, the text "Penn WebLogin" is displayed. A message reads: "Log in to gain access to many protected University web resources." There are two input fields: "PennKey Username" and "Password". Below the input fields, there is a link: "Forgot PennKey [username](#) / [password](#)". At the bottom, there is a link: "[About Penn WebLogin](#)" and a blue "Log in" button.



What goes into an app

- ✓ Persistence and data stores
Making it all matter



What goes into an app

- ✓ Integration points
Playing well with others



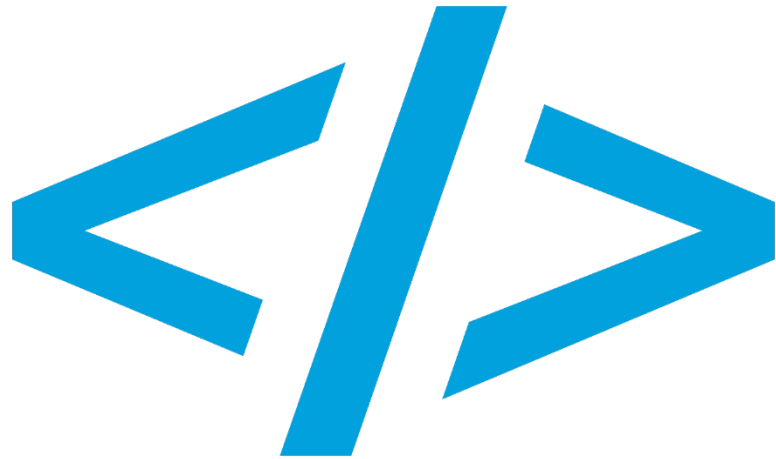
What goes into an app

- ✓ Testing
Proof it's doing what it should be



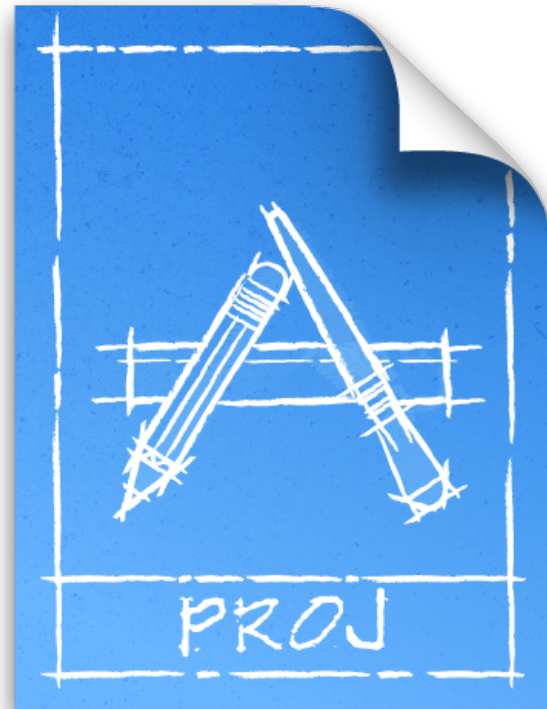
What goes into an app

- ✓ 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.

LIKELIHOOD YOU WILL GET CODE WORKING
BASED ON HOW YOU'RE SUPPOSED TO INSTALL IT:



Credit XKCD.com <https://xkcd.com/1742/>



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

26

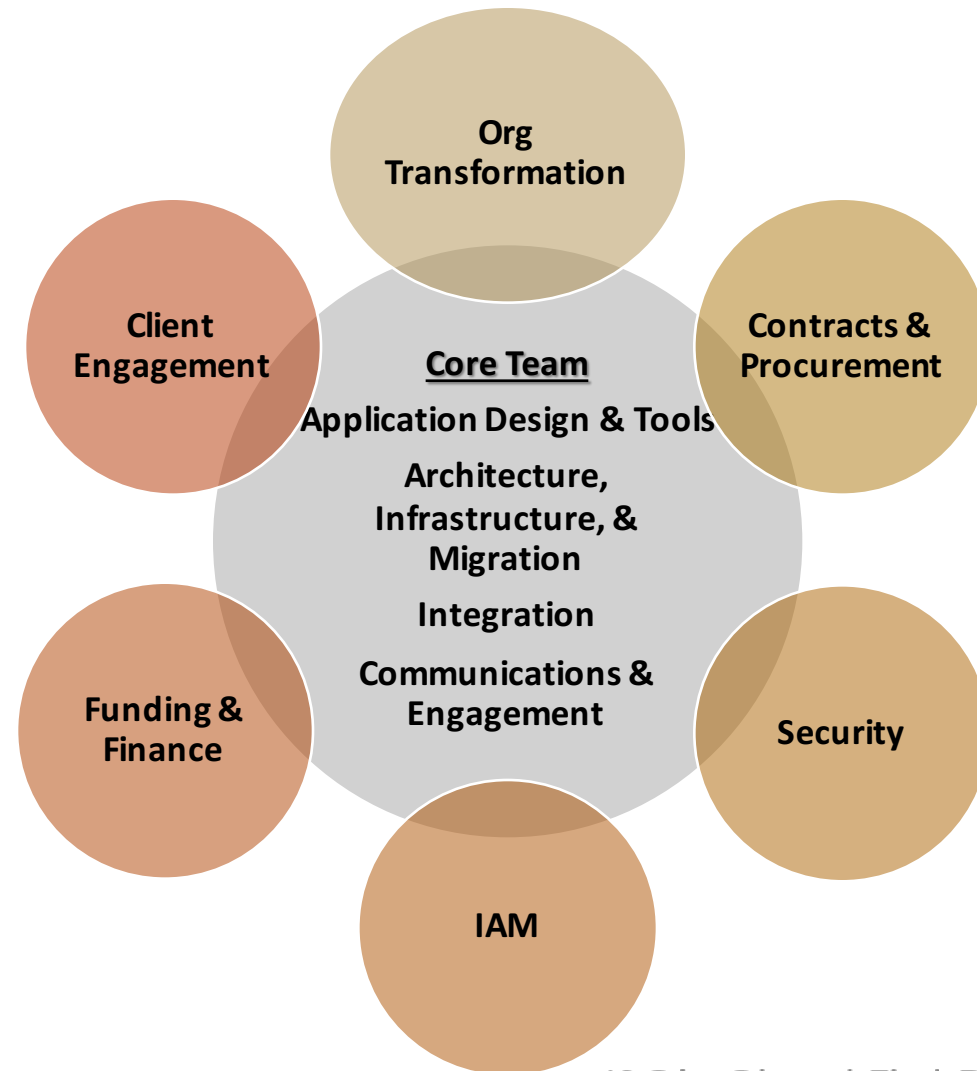
James Choate
Owner

Tiffany Hanulec
Architecture, Infrastructure & Migration

Bryan Hopkins
Application Design and Tools

Teresa Leo
Communications and Engagement

Janet Lind
Integration



ISC's Cloud First Program



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

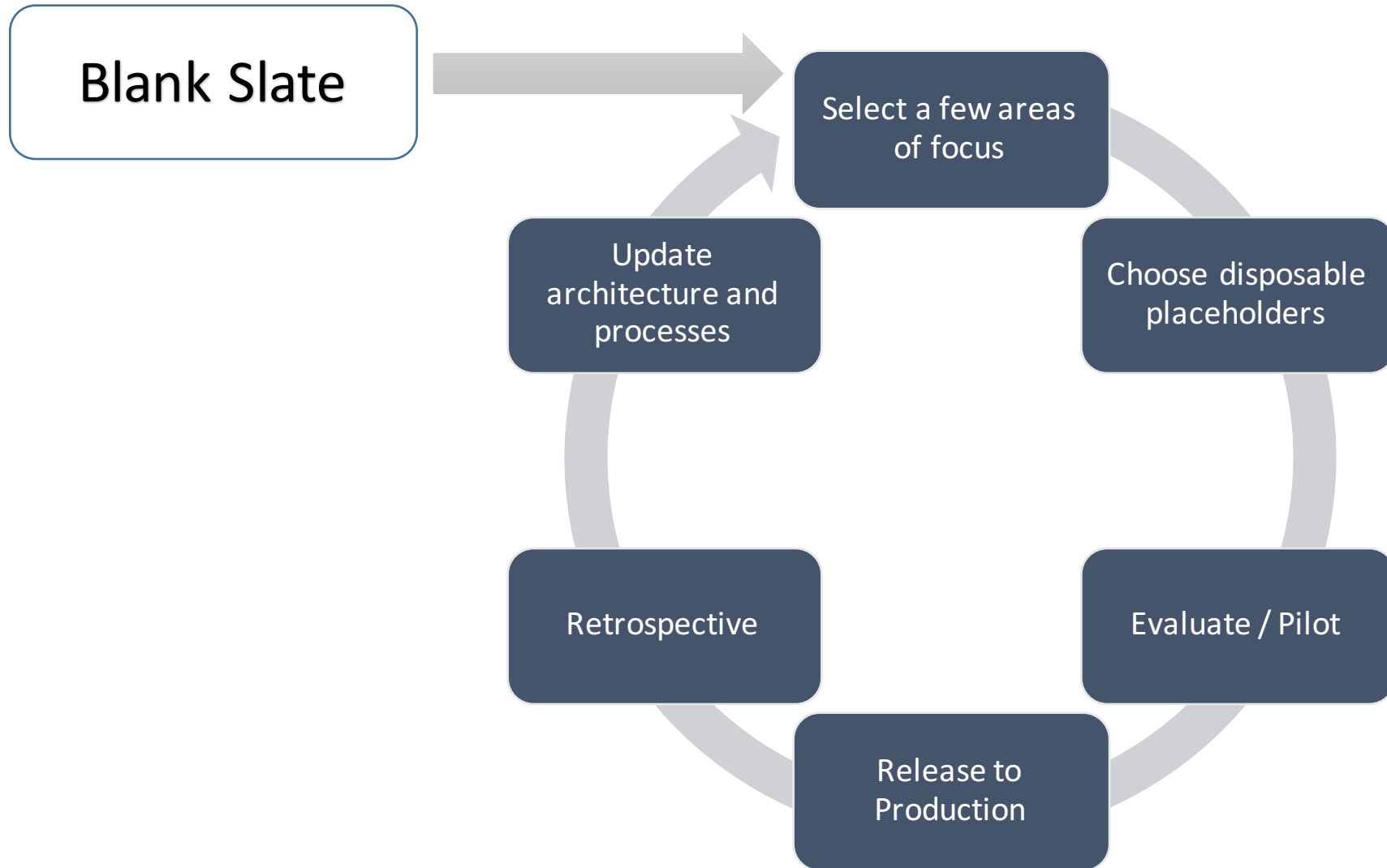


Bottom Line For All Teams...

Make us nimble integrators, and maintain low technical debt.

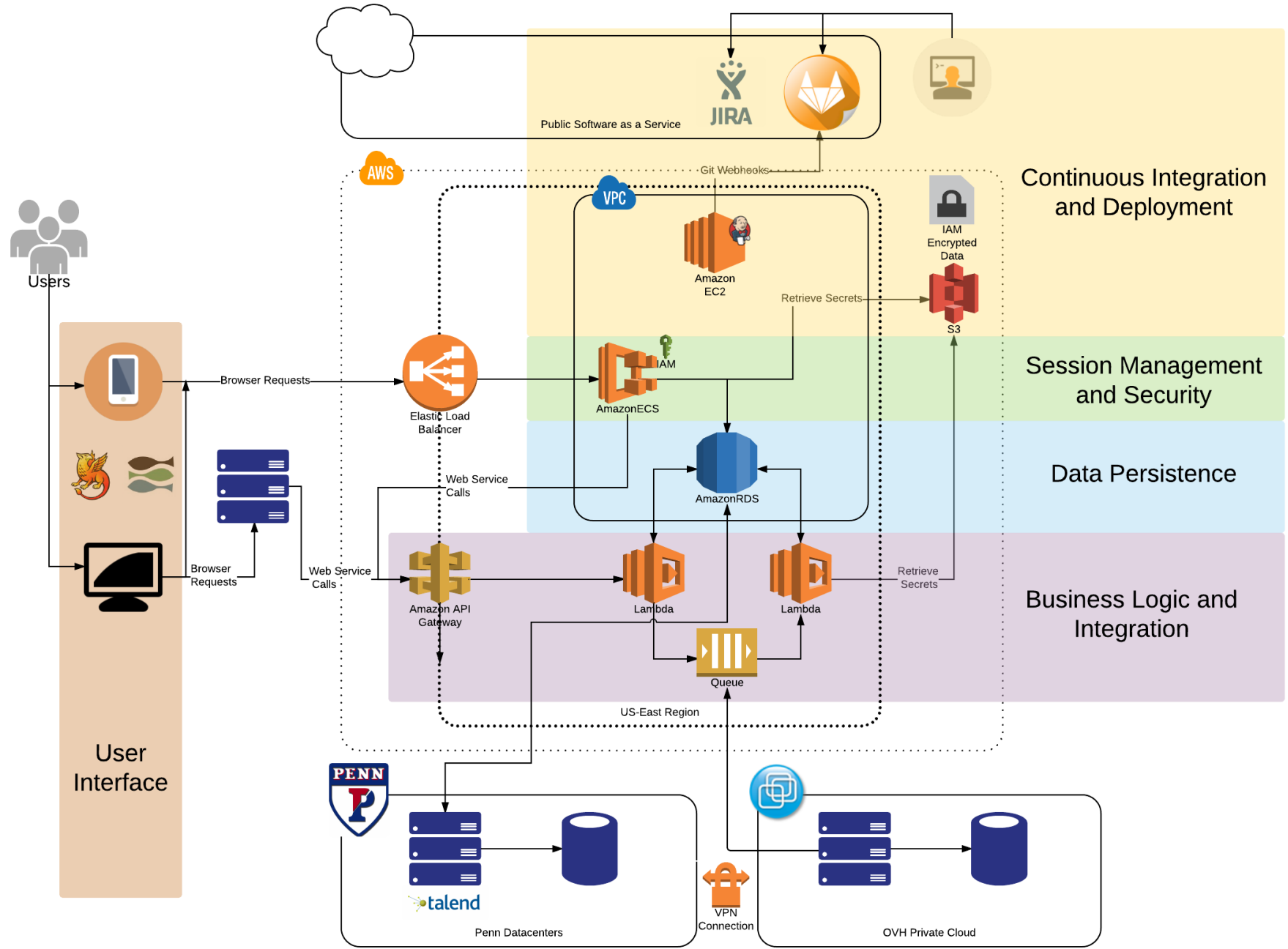


Our Process



This is a continuous process...
but where are we now?








<https://www.isc.upenn.edu/cloud-development-architecture-roadmap>



Architectural Roadmapping: More important than ever






User Interface

	For Developers...	For Analysts...	For Project Managers...
  <p>ANGULARJS Bootstrap</p>	<ul style="list-style-type: none"> • Full framework • Thousands of components • Readily available examples • Data binding 	<ul style="list-style-type: none"> • <u>Responsive</u> by default • <u>Accessible</u> components • Cleaner, simple UIs • UX vision more achievable 	<ul style="list-style-type: none"> • Open source • From Google and Twitter • Easy to find resources
 <p>JetBrains Webstorm</p>	<ul style="list-style-type: none"> • In-browser debugging • Angular-aware code completion • Variable completion with typing with JSDoc 		<ul style="list-style-type: none"> • Inexpensive • Contractors already know it






Business Logic

	For Developers...	For Analysts...	For Project Managers...
   <p>JetBrains PyCharm</p>	<ul style="list-style-type: none"> • Full framework • Thousands of packages • Readily available examples • Simple and powerful: few lines of code • Django-aware code completion • Great Django Testing integration 	<ul style="list-style-type: none"> • Self documenting APIs • Automatically generated test interfaces 	<ul style="list-style-type: none"> • Open source • Tremendous campus presence • Very easy to learn • Easy to find resources • Inexpensive with a free tier





Authentication and Security

	For Developers...	For Analysts...	For Project Managers...
   <p>JetBrains Webstorm</p>	<ul style="list-style-type: none"> • Thousands of packages • Readily available examples • Incredible I/O performance • Full stack • Framework-aware code complete • Fullstack debugging with AngularJS 	<ul style="list-style-type: none"> • No effort PennKey login with Shibboleth 	<ul style="list-style-type: none"> • Open source • Easy to find resources • Inexpensive with a free tier





Data Store

	For Developers...	For Analysts...	For Project Managers...
 <p>PostgreSQL</p>	<ul style="list-style-type: none"> • Oracle-like features • That actually work! • Highly performant • Relational AND NoSQL 	<ul style="list-style-type: none"> • Drop-in replacement for Oracle • Works great with Talend • DB of choice for warehousing 	<ul style="list-style-type: none"> • Open source, ethical, and Free • Strong local presence
 <p>JetBrains DataGrip</p>	<ul style="list-style-type: none"> • Works with every RDBMS • Powerful autocompletion 	<ul style="list-style-type: none"> • Very easy to use • Don't need to know SQL 	<ul style="list-style-type: none"> • Inexpensive



Integration

	For Developers...	For Analysts...	For Project Managers...
 <p>API Gateway</p>	<ul style="list-style-type: none"> • Fully supported SDK in every language • Easy to follow examples • Framework and platform agnostic • Built in DDoS protection 	<ul style="list-style-type: none"> • Powerful and flexible security models • API Keys and Usage plans • Configurable “stages” 	<ul style="list-style-type: none"> • Incredibly cheap • Fully managed • Serverless • Taggable for <u>cost transparency</u> • Service orientation
 <p>Amazon SQS</p>	<ul style="list-style-type: none"> • Fully supported SDK in every language • Failure tolerant • Limitless scalability 	<ul style="list-style-type: none"> • Works with Mule 	<ul style="list-style-type: none"> • Inexpensive • Taggable for <u>cost transparency</u>



Integration



For Developers...

- Integrates directly with AWS Services like S3 and RDS
- Fully automated

For Analysts...

- No code to write or support
- Near real-time replication (5 minutes)

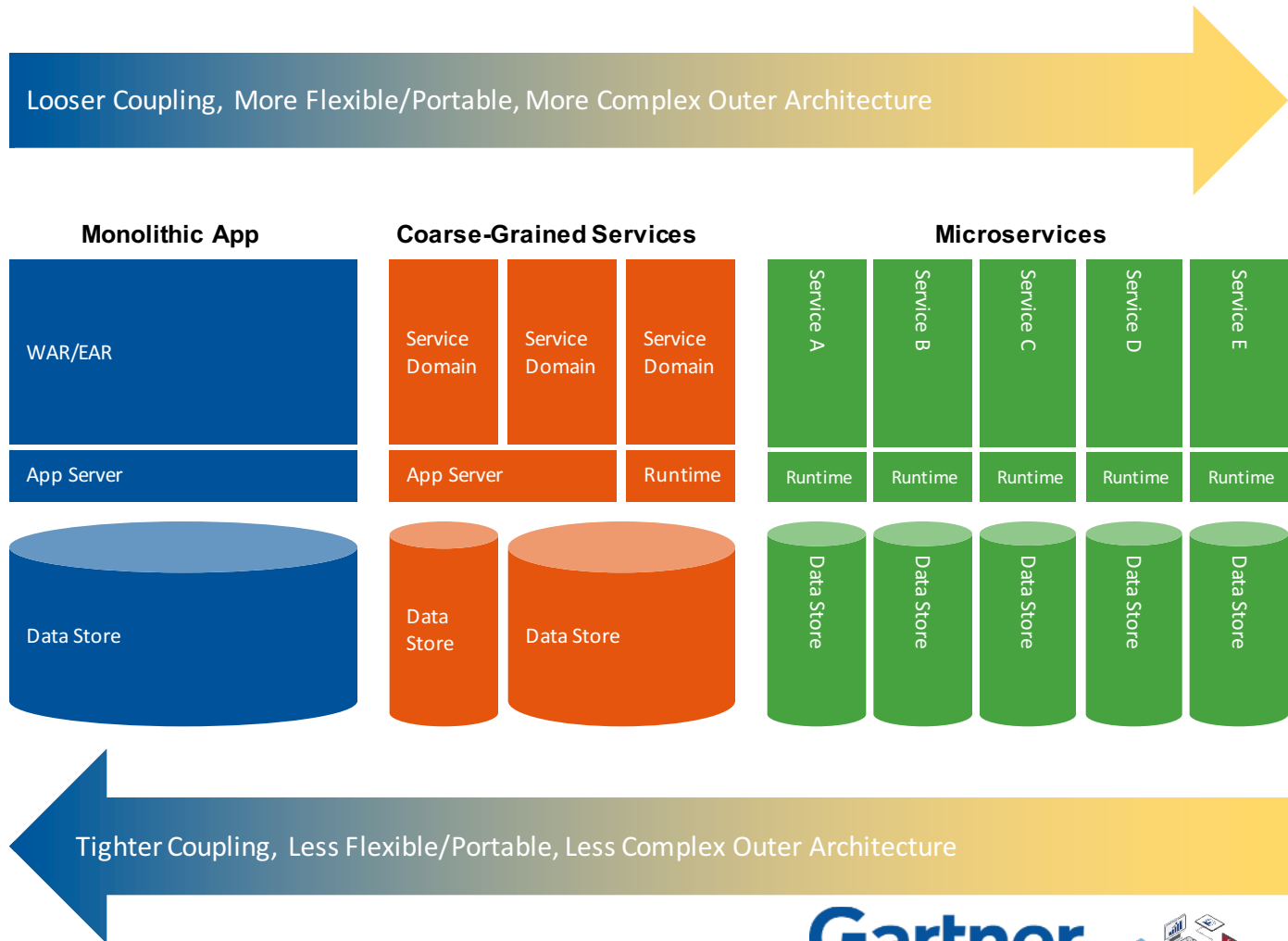
For Project Managers...

- Managed Software as a Service
- Great support and community



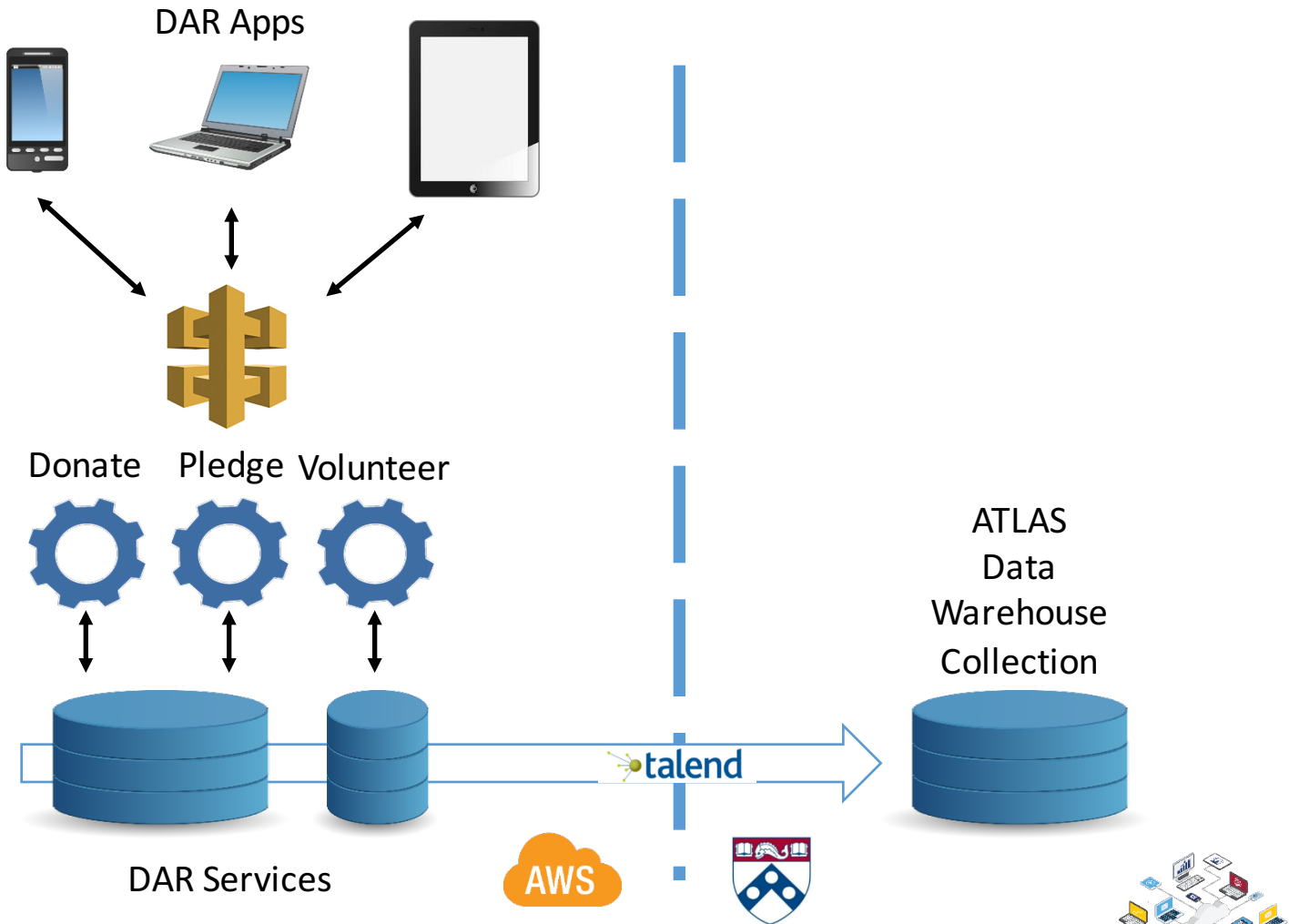
Closer Look: Microservices and SOA

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







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

	For Developers...	For Analysts...	For Project Managers...
 	<ul style="list-style-type: none"> • Confidence in the spec 	<ul style="list-style-type: none"> • Easy to create and understand • Fully automated regression tests 	<ul style="list-style-type: none"> • Bugs caught early take 10x less time
 <p>KarmaJS</p>	<ul style="list-style-type: none"> • Built for AngularJS • Easy stubbing and mocking with Sinon 		
	<ul style="list-style-type: none"> • Straightforward • Like Junit • <i>Easy</i> to test • WebServices: no setup 		



Source Repository

For Developers...

- Offline work
- Easy branches and merging
- Webhooks for CI/CD integration

For Analysts...

- Create documentation developers actually use
- Intuitive web interface to browse files

For Project Managers...

- Open source
- Universal adoption (git)



Gitlab Cloud



Deployment Pipeline

For Developers...

- Automate with Python
- Developers' Swiss Army knife

For Analysts...

- More frequent deployment
- Automated tests run on each build
- Slack Integration

For Project Managers...



Jenkins



Platform

For Developers...

For Analysts...

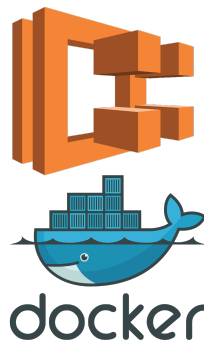
For Project Managers...



Amazon Lambda

- Serverless
- [Only worry about code](#)
- Provisioned as code
- Isolated microservices
- Easy DRF integration with Zappa

- Incredibly cheap
- Nothing to patch
- Massively scalable
- Taggable for cost transparency




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

- Can be sized and scaled to order
- Taggable for cost transparency



Platform

	For Developers...	For Analysts...	For Project Managers...
 Amazon RDS	<ul style="list-style-type: none">• Fully Managed Service• Provisioned as code	<ul style="list-style-type: none">• Access like any other database	<ul style="list-style-type: none">• Fully managed• Nothing to patch• Scalable• Taggable for <u>cost transparency</u>



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"  
}
```



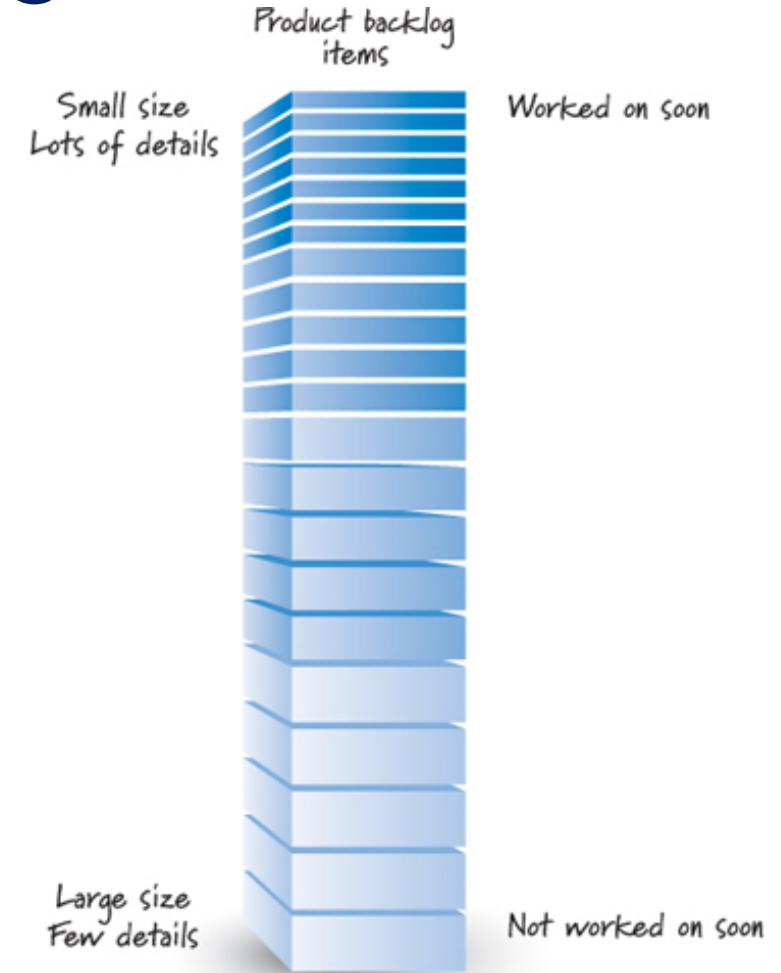
If everything is code, process from that
world matters more



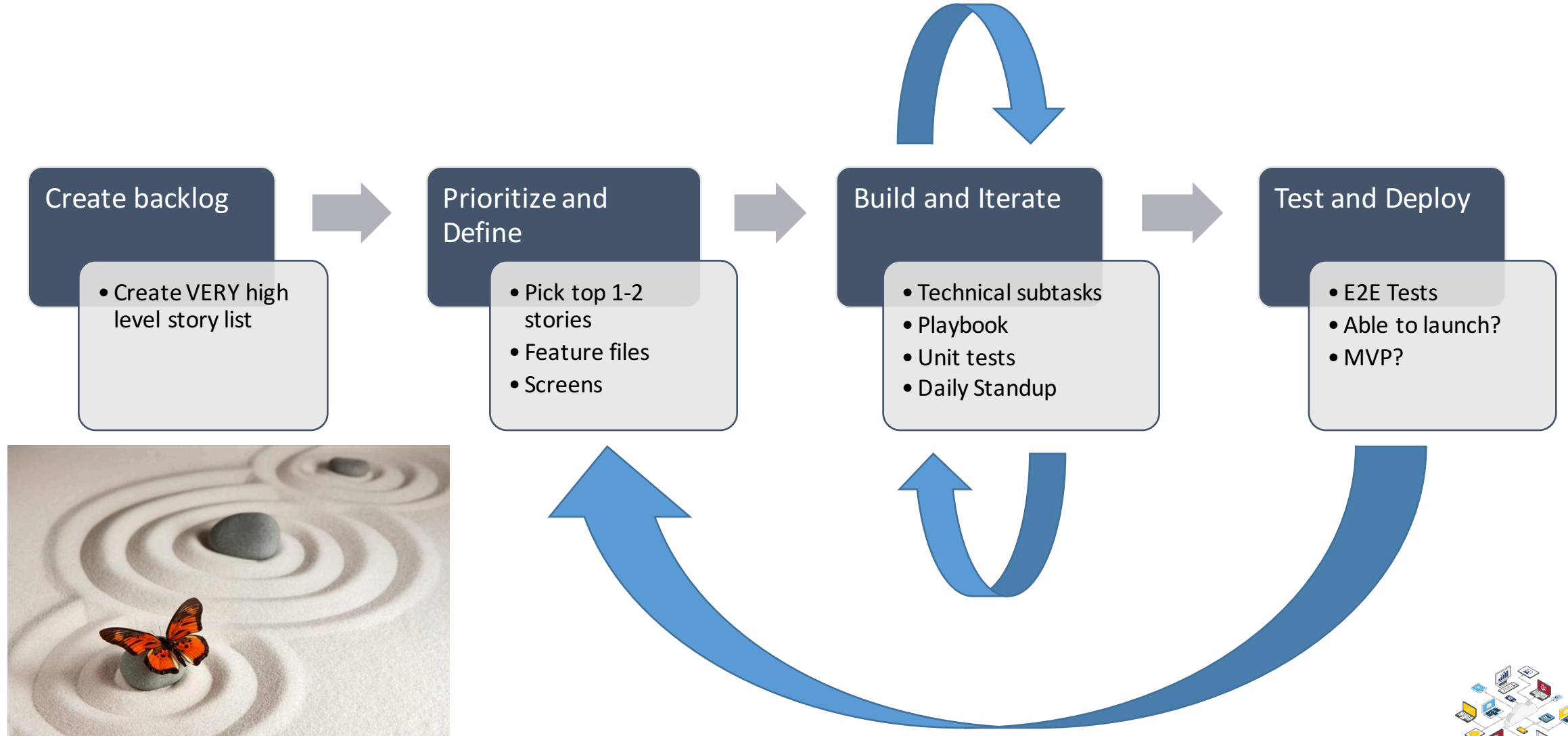
Closer Look: Cloud First Agile

Terminology


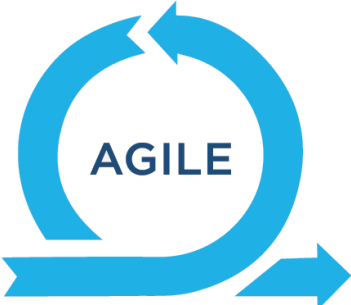
- Backlog
Sorted list of everything the clients want.
Things at the top are better defined than things at the bottom.
- Sprint
Period of time where the team focuses on implementing the clients' top priorities.
- User story
The basic unit of work capturing a feature or requirement from the end user's perspective.
- Epic
A collection of User Stories that together define a significant feature or business process



Closer Look: Cloud First Agile



Process and SDLC

	For Developers...	For Analysts...	For Project Managers...
	<ul style="list-style-type: none"> • Familiar tool • Easy to use • Highly collaborative 	<ul style="list-style-type: none"> • Plugin store • Dependency tracking • Drag and drop prioritization 	<ul style="list-style-type: none"> • Transparency • Reports and progress-at-a-glance • Collaborate externally
	<ul style="list-style-type: none"> • Focus on only a few details at a time • Frequent feedback and adjustment 	<ul style="list-style-type: none"> • Focus is on “Yes” • Rely on domain knowledge and organization • No document writing 	<ul style="list-style-type: none"> • <u>Speed</u> and time-to-value • Responsive to changes in priorities



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



Credit: The Register https://www.theregister.co.uk/2017/03/01/aws_s3_outage/

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

is shib hosed? wasn't aware of that, checking

its fine

afaik we are 100% unaffected

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

