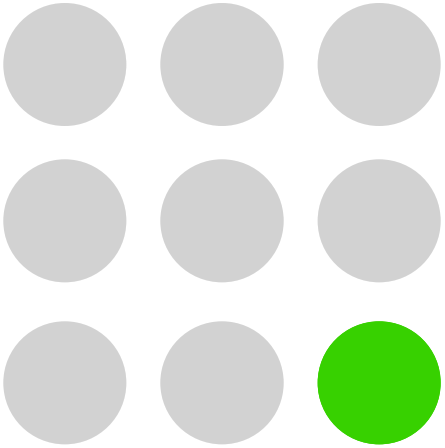
Testing in the Cloud.

Not just for the Birds, but Monkeys Too!

PhUSE Conference
Paper AD02

Geoff Low | Software Architect I

15 Oct 2012

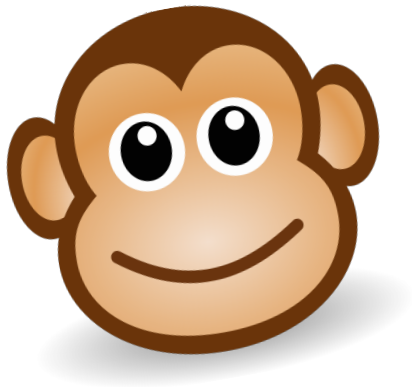


medidata®
Medidata Solutions Worldwide

Optimising Clinical Trials:
Concept to Conclusion™

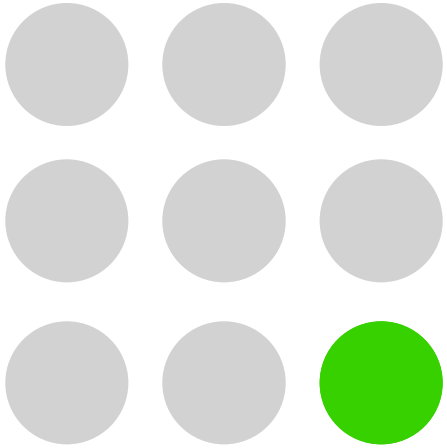
The flow….

- What is the cloud?
- Testing Paradigms – Continuous Integration
- Testing in the cloud
- The Chaos Monkey



medidata
Medidata Solutions Worldwide

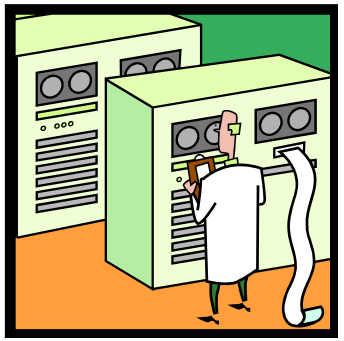
What is the cloud?



medidata®
Medidata Solutions Worldwide

The Cloud

- The cloud treats computing resource like a "commodity"
  - Storage
  - Processing

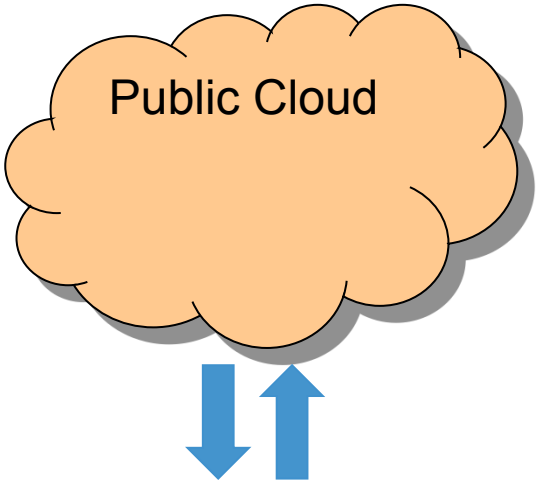


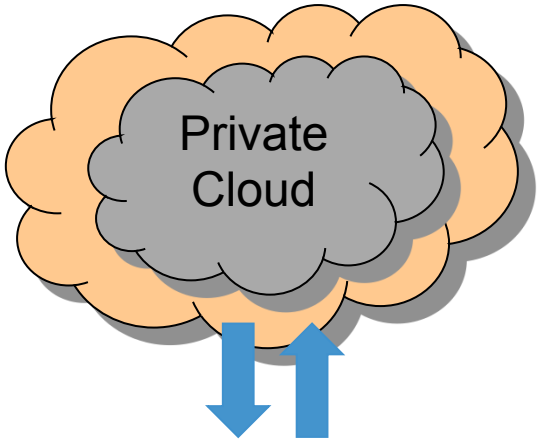
- Seems like a relatively new concept
  - Concept originated in '60s
  - Time-slicing – selling access to a processor

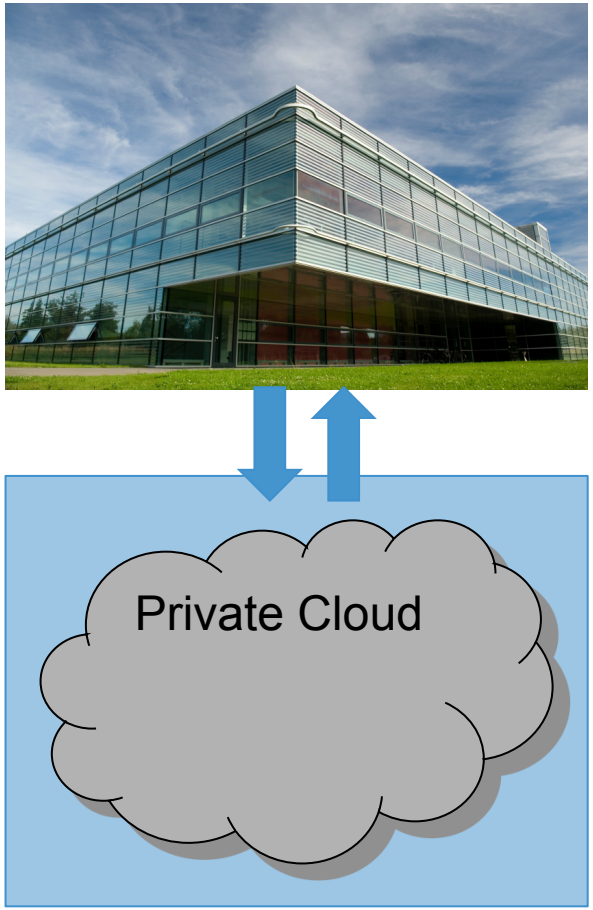
- Commoditization
  - Providers have large datacenters with trained and certified staff (who regularly deal with 'sensitive' data)
  - Sell off extra processing power to people

medidata
Medidata Solutions Worldwide

Clouds – Public and Private



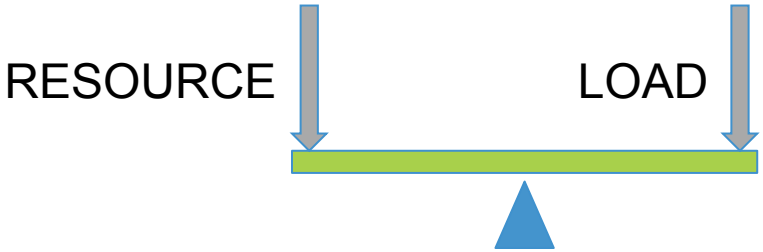




medidata
Medidata Solutions Worldwide

Power of the cloud - Auto-scaling

- Known - Pooling of server types
  - Using tiers (eg Application Server, Web Server)
  - Define a number of servers at build time to cope with expected load
  - Add servers as required

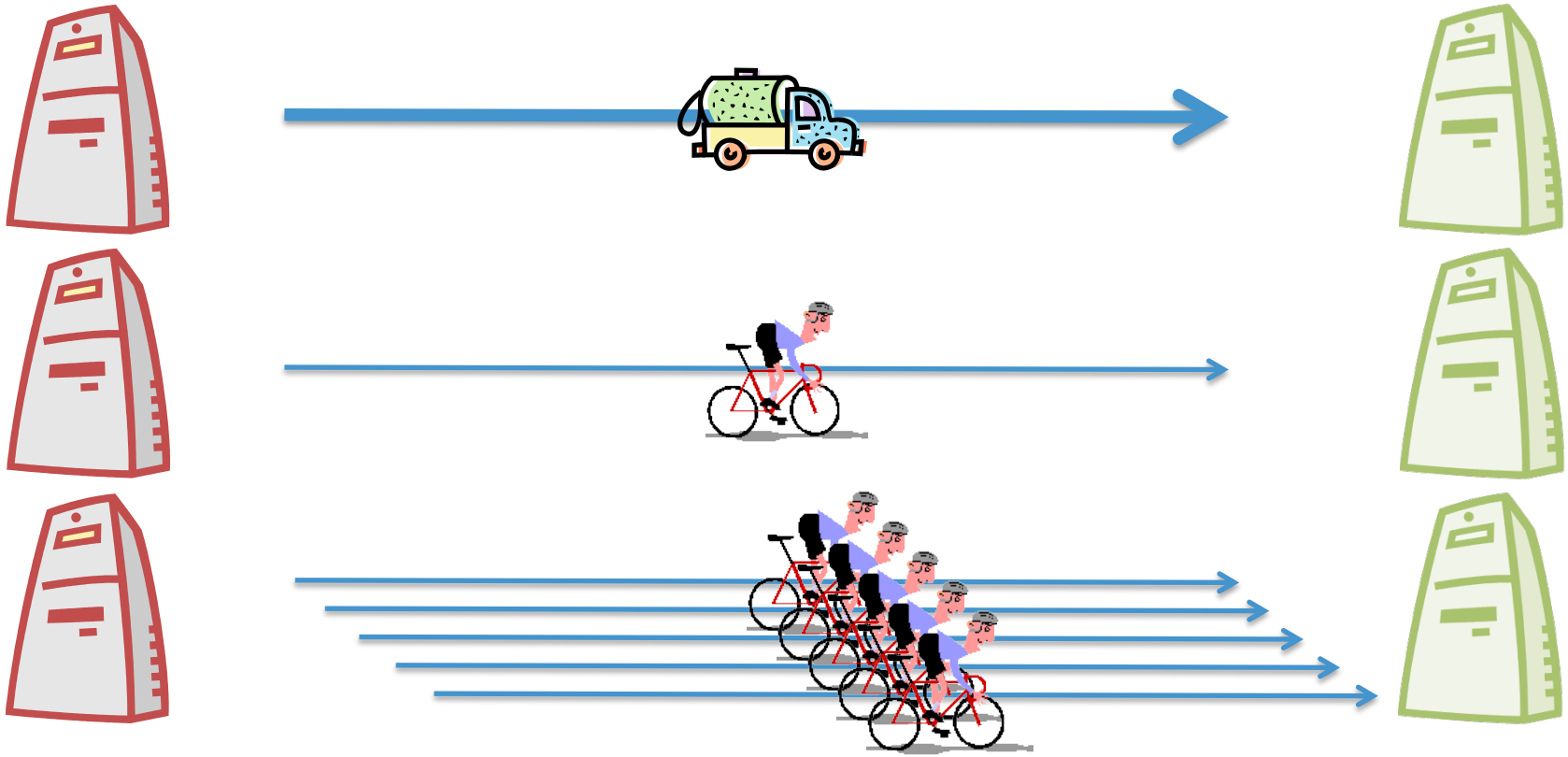


- Auto-scaling automates this
  - High load, servers automatically added to a pool
  - Lower load, servers automatically taken from the pool

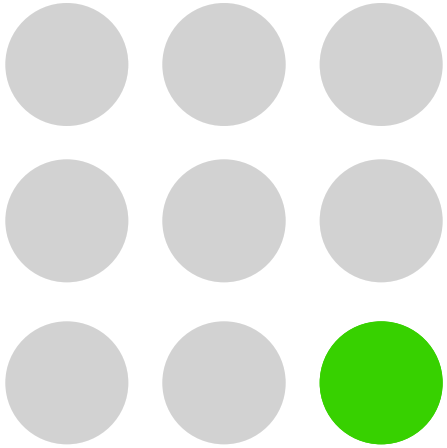
- Defining ‘load’
  - Metrics about application performance (eg ping, Apdex)

medidata
Medidata Solutions Worldwide

An analogy - transporter



Testing Paradigms – Continuous Integration



medidata®
Medidata Solutions Worldwide

Continuous Integration Testing

- Continuous Integration Testing
  - Moving QC processes to much earlier on in the development lifecycle
  - Running Build and Test cycles up front
  - Running Tests on an ongoing basis

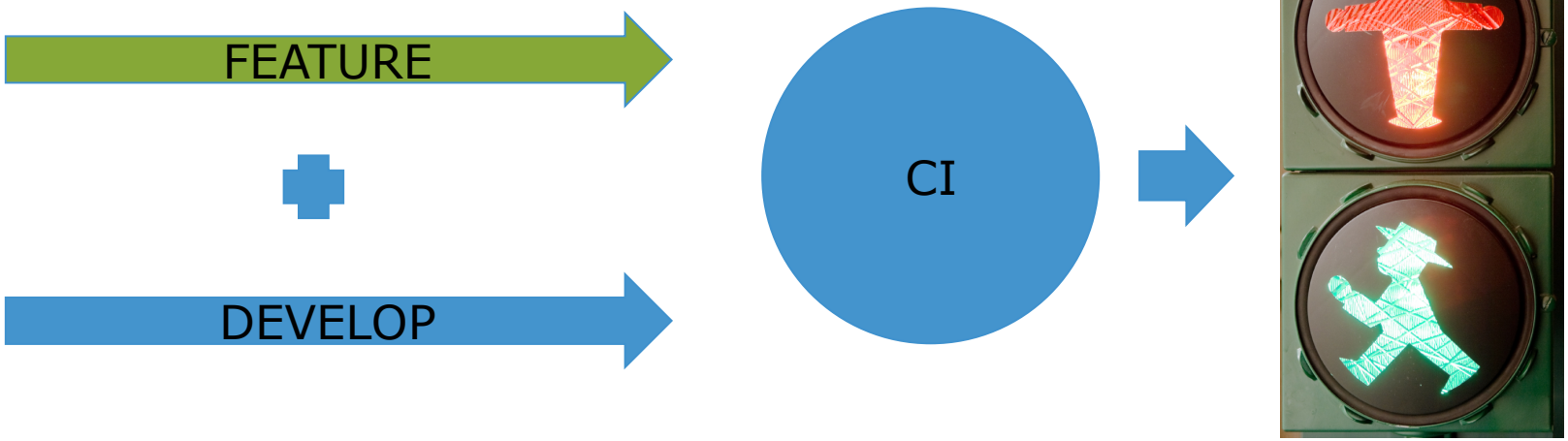
- Important part of an Agile development methodology
  - Many streams of work ongoing in parallel
  - Numerous changes to the source code
  - Each of these feature branches will need to be merged into the develop branch
  - Continuous integration of development branch

medidata
Medidata Solutions Worldwide

CI Best Practise

- Have the (develop + feature) branches in CI Testing
  - as well as the current develop branch

- Reduces 'merge and SURPRISE' issues

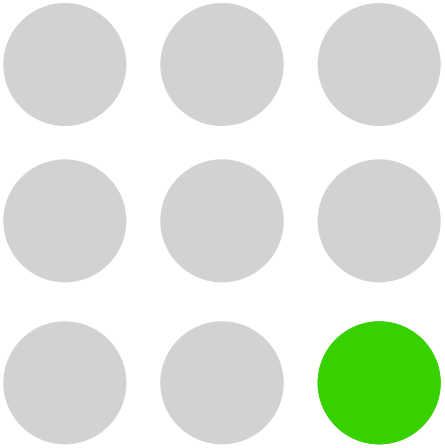


Implementation

- A common pattern with Jenkins
  - CI is carried out with the head of a feature branch merged into develop – with every commit on a feature branch
- The system revolves around GitHub and Jenkins
  - Jenkins installs 'hooks' on GitHub repositories
  - On a write or pseudo-write action, a callback is made to the CI server
  - The CI server pulls the code from GitHub, builds, runs tests and reports the status (build or test pass|fail)
- Issues we have
  - Need to be able to flag commits as WIP to prevent binding up a testing machine with changes we know might break CI

medidata
Medidata Solutions Worldwide

Testing using the Cloud



medidata®
Medidata Solutions Worldwide

Continuous Integration is Important

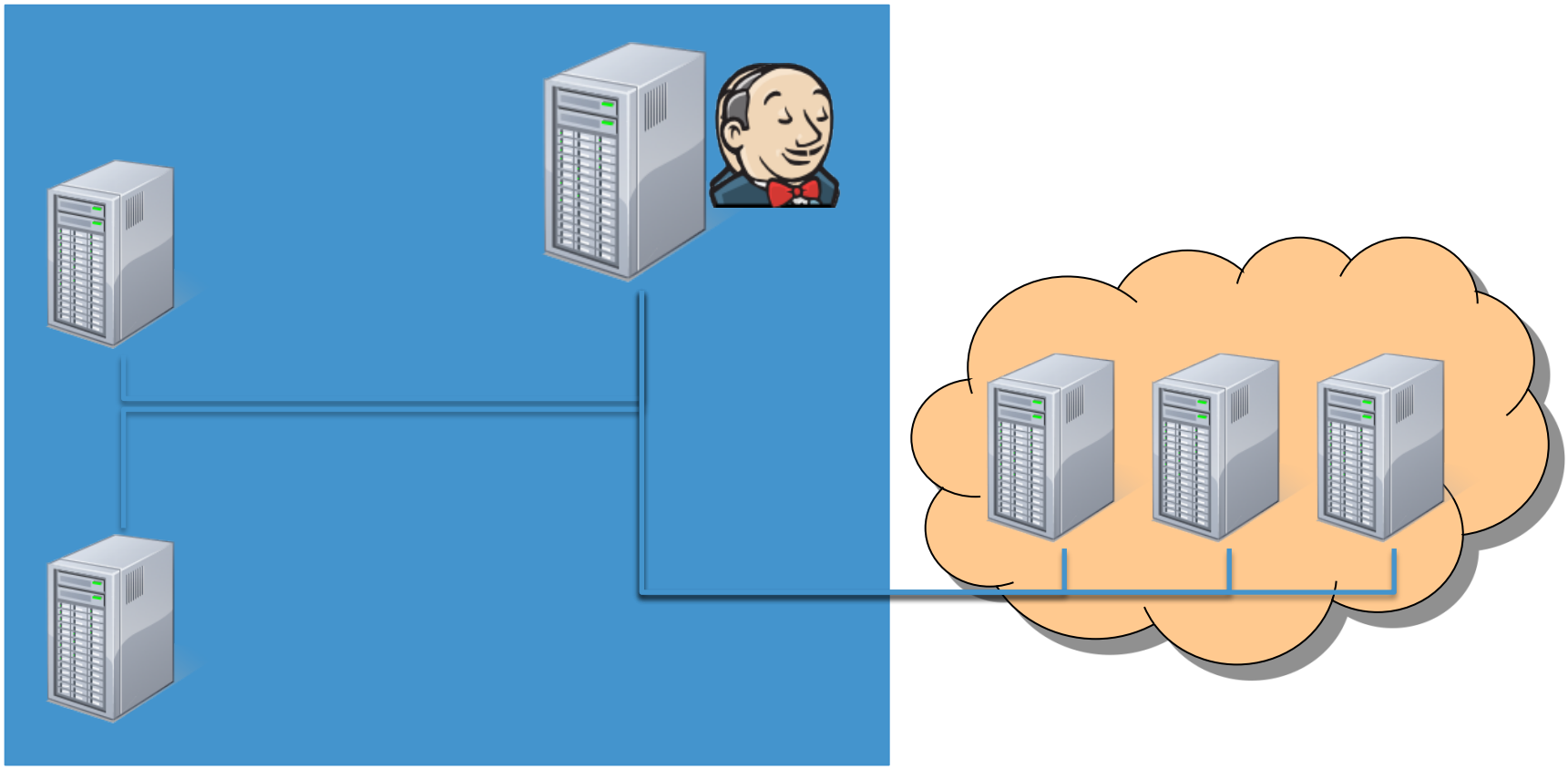
- We make it a fundamental part of our development

- Ongoing Development in parallel
  - Competition for testing resources!
- Continuous Integration server builds the software and then farm it off to a slave server to execute tests
  - Limited to a certain number of slaves, by number of test servers that are available
  - Can build up 'detritus' over time – tests take longer to run
  - Developer downtime has a large effect on sprint planning and execution

medidata
Medidata Solutions Worldwide

Expandable Continuous Integration System

- The Continuous Integration Server slaves out to testing instances
  - Use a Cloud testing infrastructure – defined test instances.



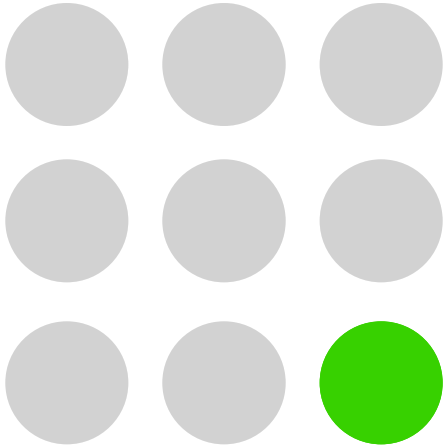
medidata
Medidata Solutions Worldwide

Best of both worlds!

- Latency for cloud instance is higher
  - Put it after the local servers
- No upper limit on the instances
  - How much do you want to pay for?
  - Doesn’t cost anything when not required.
- Same mechanism for both instances for deployment and test
- Problems:
  - Doesn’t currently work with Windows Instances

medidata
Medidata Solutions Worldwide

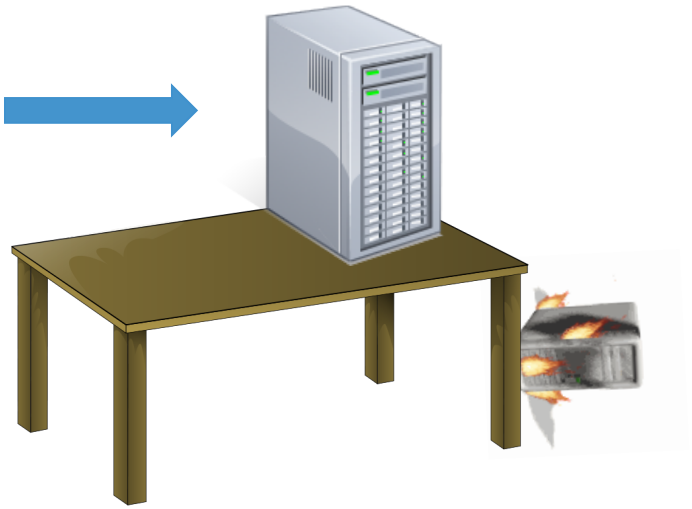
The Chaos Monkey



medidata®
Medidata Solutions Worldwide

The Chaos Monkey

- Developing system resilience is an important part of testing
  - How to test resilience?



- This is one way
  - We'd like it to be a bit more reproducible (and less expensive)

- "the positive ability of a system or company to adapt itself to the consequences of a catastrophic failure"

medidata
Medidata Solutions Worldwide

Netflix

- A well-known adopter of the cloud for services delivery – over 1 Petabyte of data in the cloud

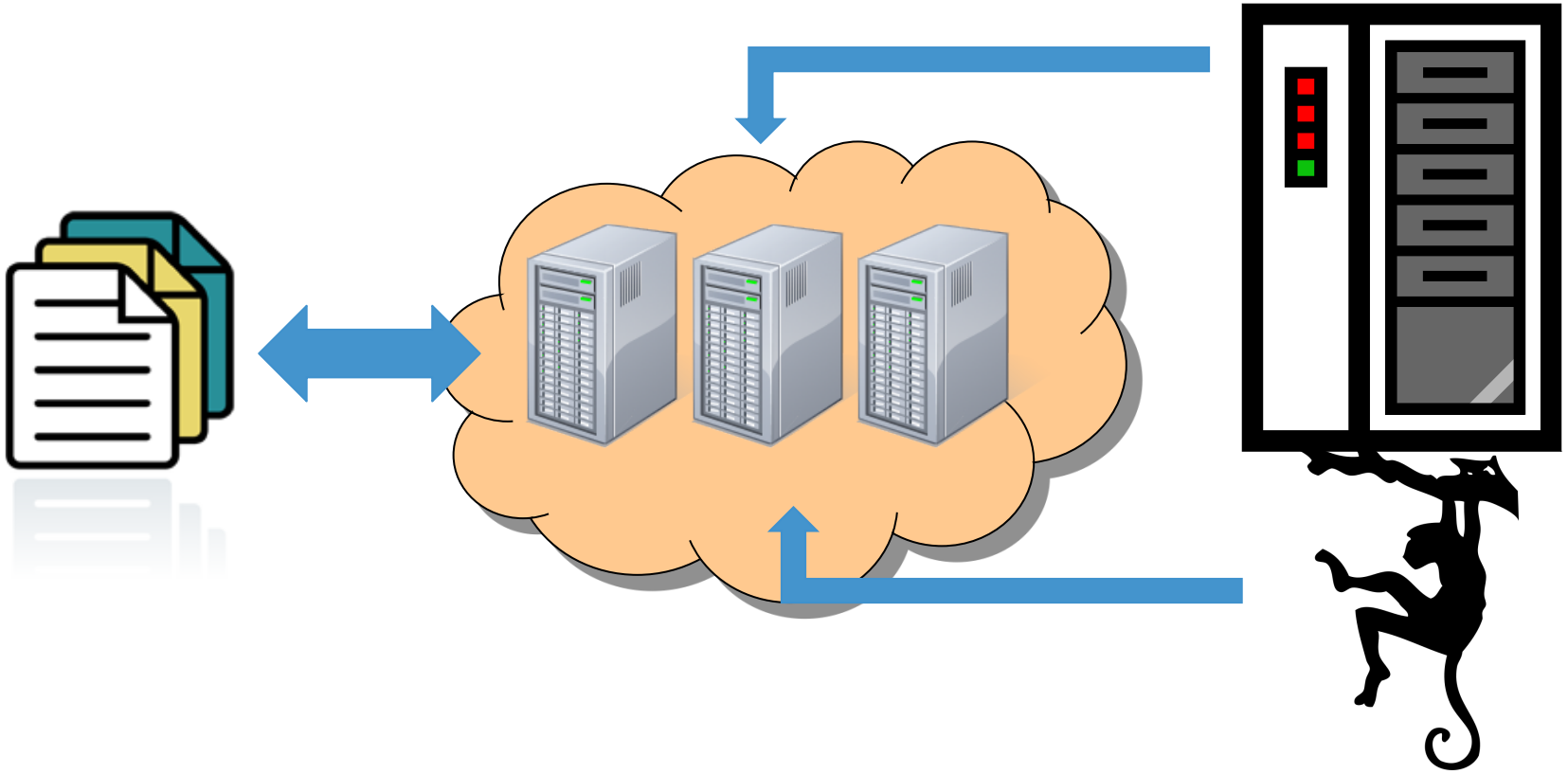
- A multi-tier system with many services
  - Authentication
  - Recommendations
  - Movie Data
  - Member Date

- All hosted on AWS
  - Discovery services built in
  - Auto-scaling Groups (ASGs)
  - Massive DB installations
  - Using a service called Eureka to manage middleware

medidata
Medidata Solutions Worldwide

The Chaos Monkey

- In order to ensure that services don't break uncontrollably when they are out of the office, they intentionally break them (between the hours of 9-5)



medidata
Medidata Solutions Worldwide

Failure as an component

- We can’t do this as part of a production environment
  - Services are much less numerous and therefore much more valuable
- We can incorporate the concept of planned failure into a testing regime.
  - Normal testing - build, deploy, run tests, tear down
  - Chaos testing – build, deploy, “break stuff”, run tests, tear down
- Netflix have expanded their ‘tribe’
  - Janitor Monkey
  - Latency Monkey

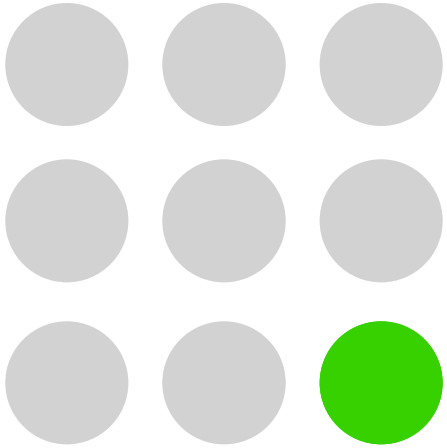
medidata
Medidata Solutions Worldwide

Where we're at!

- Defining a strategy for platform deployment
  - Moving from n x Products to 1 x Platform
- Defining failure cases
  - Turn these into Monkeys (eg Cache Failure Monkey)
- Reviewing open-sourced Simian Army
  - (check GitHub – another cloud based services delivery agent)
- Combine all services into a platform test
  - With build-in failure included!

medidata
Medidata Solutions Worldwide

Conclusion



medidata®
Medidata Solutions Worldwide

Testing in the Cloud

- The cloud presents almost limitless capacity for testing and "trying things out".
  - Not limited by hardware you have to hand
- Very powerful tools exist
  - AWS EC2 API, etc
  - Opscode Chef – deployment platform
- Leverage these
  - Cut down time spent in administration
  - More time can be spent testing (in parallel) with fewer queues for resources

medidata
Medidata Solutions Worldwide

