Automation w/ Puppet & a Path to Private Hybrid Cloud

by Andrew Ludwar

Lead, UNIX/Linux Infrastructure

Agenda

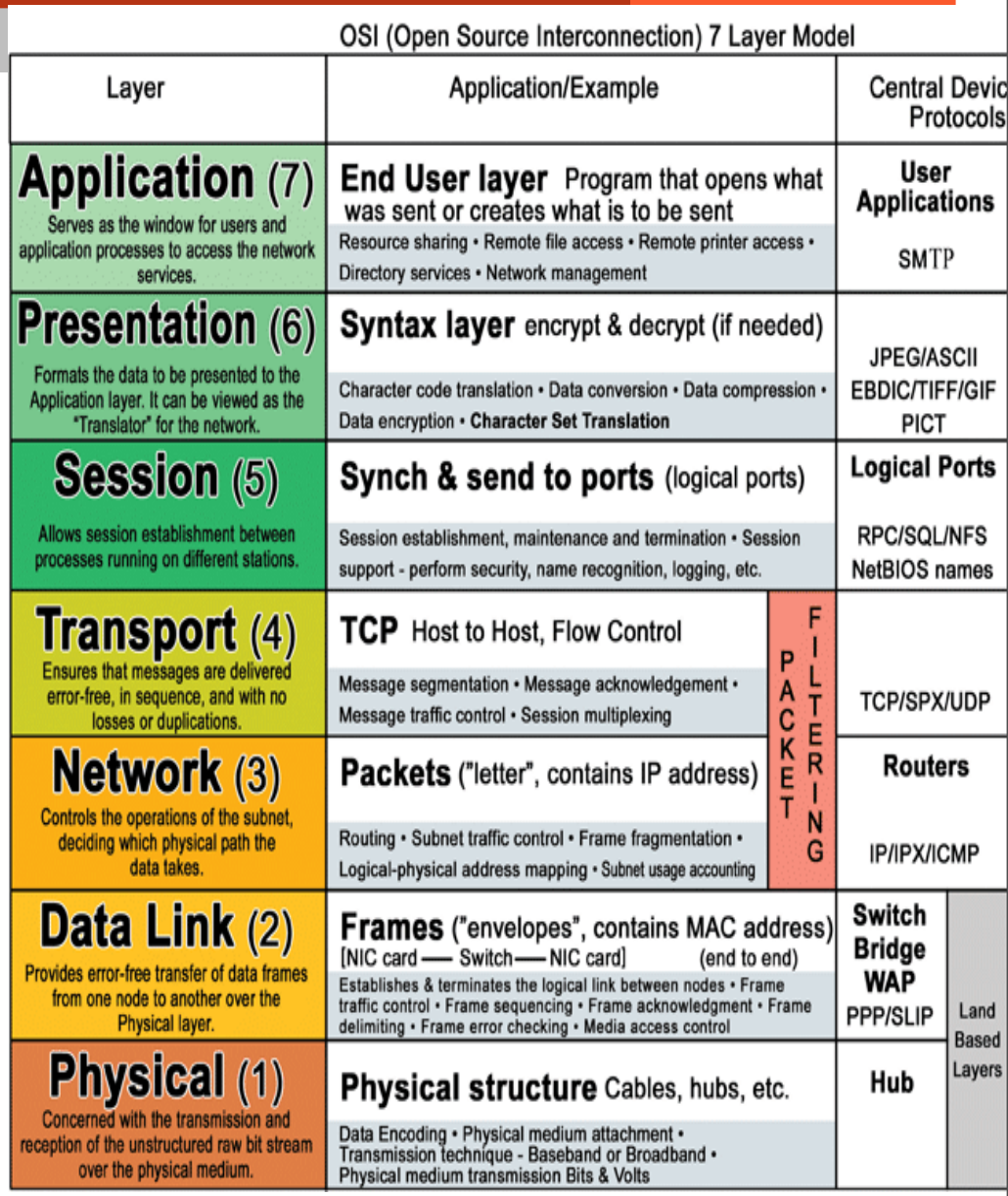
- Automation
  - What is Puppet?
  - Where does it apply?
  - Gotcha's / best practices
- Path to Private Hybrid Cloud
  - Server provisioning

What is Puppet?

- Automation? – … in the right context, yes!
- Orchestration? – No!
- Configuration Management? – Yes!
- Fancy scripting? – Yes!
- Marketing fluff? – Couldn’t be more true!

What is Puppet?

- Automation… let's define it further
  - Drift management for server/VM "sprawl"
  - Repeatable, reliable provisioning
  - Programmatical approach to infrastructure
- What it's NOT
  - Orchestration language
  - Silver bullet to automation
  - Exists below layer 3 in OSI



What is Puppet?

- Configuration management tool & scripting language
- State Described Language (SDL)
- Client/server architecture

  puppet client agent --> puppet master server
- Uses SSL certificates; encrypted communication
- OS Platform agnostic (**mostly**, true for Linux, Solaris/Windows not quite as mature)
- Abstracted syntax

  apache installed = true; instead of

  yum install apache && chkconfig apache on && service apache start
  apt-get apache, etc.

What is Puppet?

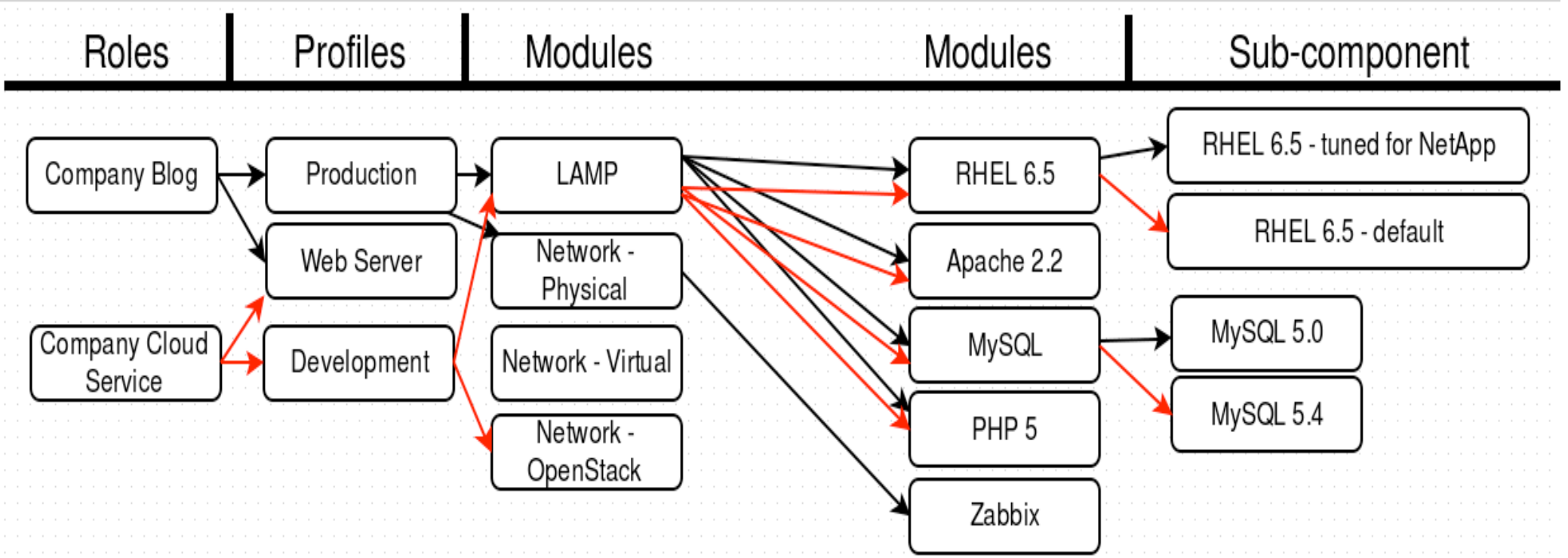
- Reintroduces traditional modular programming concepts into “Puppet Modules” (functions, methods)
- Maintains state of servers based on:
  - Server/OS facts collected into a catalog called Factor/Heira
  - Unique identifiers of hosts (FQDN usually)
  - Node groups and other classification techniques (ENC – External Node Classifier)
  - Hardcoded classification and modules (will haunt you eventually)
- Assumes server is on the network (TCP/IP, not TFTP/BOOTP)
- Agents check in typically every 15 minutes
- Config changes should follow a CI/CD workflow

What is Puppet?

- Enterprise & open source versions for both puppet software and the puppet code written with it
- Written by developers for developers (enterprise consumable?)
- Large reason for the uptake in the “DevOps” culture and processes
- Becoming highly adopted & integrated in the open source ecosystem (& some proprietary)
  - RedHat Satellite 6, RedHat OpenStack, OpenShift, etc.
  - Canonical Landscape, JuJu, MaaS, etc.
  - VMware vCloudAC, Amazon EC2, Google Compute, F5, Cisco, etc.

Managing Puppet

- Roles --> Profiles --> Modules --> Sub-components
  - A node can only have one role at a time (think in terms of the business)
  - A role includes one or more profiles to define that type of server
  - A profile includes and manages modules to define a logical technical stack
  - Modules manage resources and should only be responsible for managing aspects of the component they are written for



What does Puppet mean for me?

- Am I going to have to learn a new scripting language?
  - Potentially, but the syntax is very English & UNIX like

    file_permissions { root : root };
    apache_installed = true; apache_web_root = '/var/www/html'

    - Bulk of work is classifying systems, identical to kickstart/ jumpstart. You wouldn't necessarily have to write the kickstart code, but you should be able to kick a server.
- Is automation going to put me out of a job?
  - Not going to happen. Less time is spent on low-value tasks, and more time being in control of your systems.
  - Automation has been around for years
  - Experience is key

Experience is key…

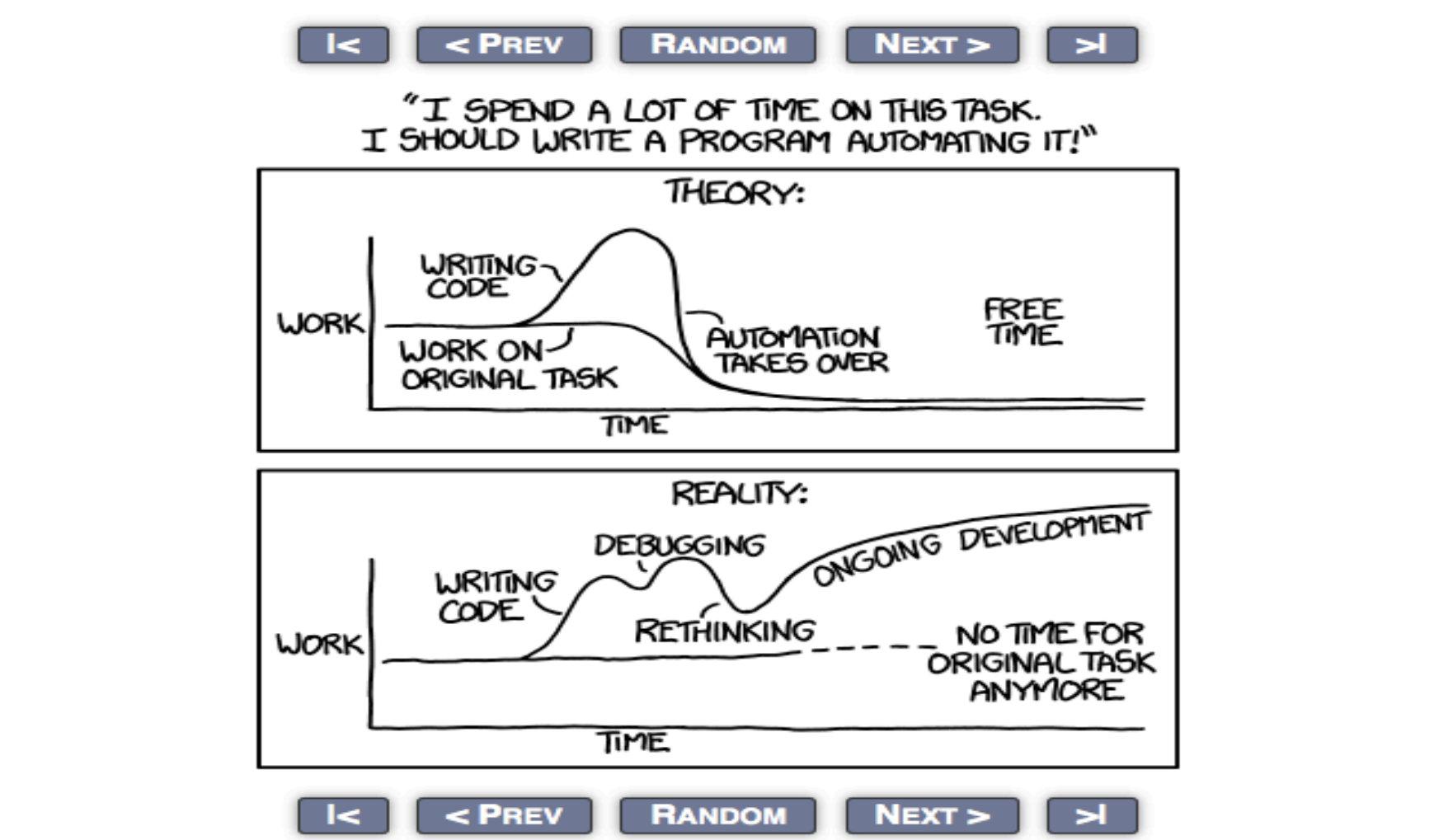
Ted, the factory manager, was having difficulty in his assembly line. Ted hired his friend Brian, a senior engineer, to see if he could help identify the problem area. Brian walked up to the control panel, watched the line for 10 minutes, pressed one button, and resolved the issue. Ted was thrilled, and told him to send an invoice.

The bill arrived, for $10,000. Ted, a little shocked, asked for a breakdown. Brian sent another invoice, indicating a $1 charge for pressing the button, and $9,999 for knowing what button to push.

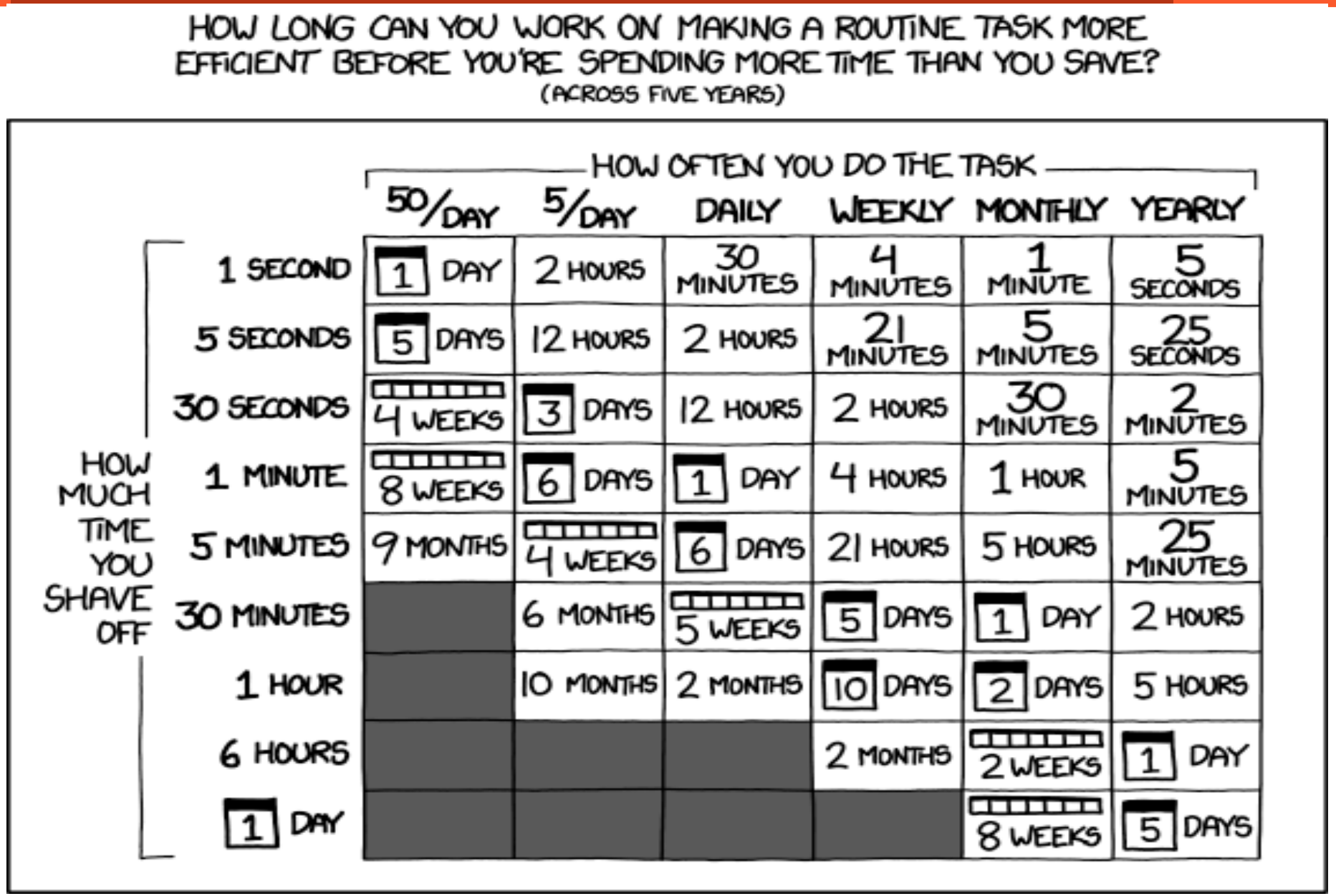
What makes sense for your company?

- Puppet enterprise, puppet open source, or leverage Satellite?
- Integration with other tools? (Microsoft System Center?)
- Enterprise tested modules from vendor or roll your own? Hybrid?
- What % of company resources spent being administrators vs developers?
- Code maintenance costs? Future portability?
- Integration with other teams?

What to be careful of?



What to be careful of?



What to be careful of?

- Indecision comes at a high cost
- Think courageous failures!
- Unconventional thinking

"In a time of rapid change, standing still is the most dangerous course of action."

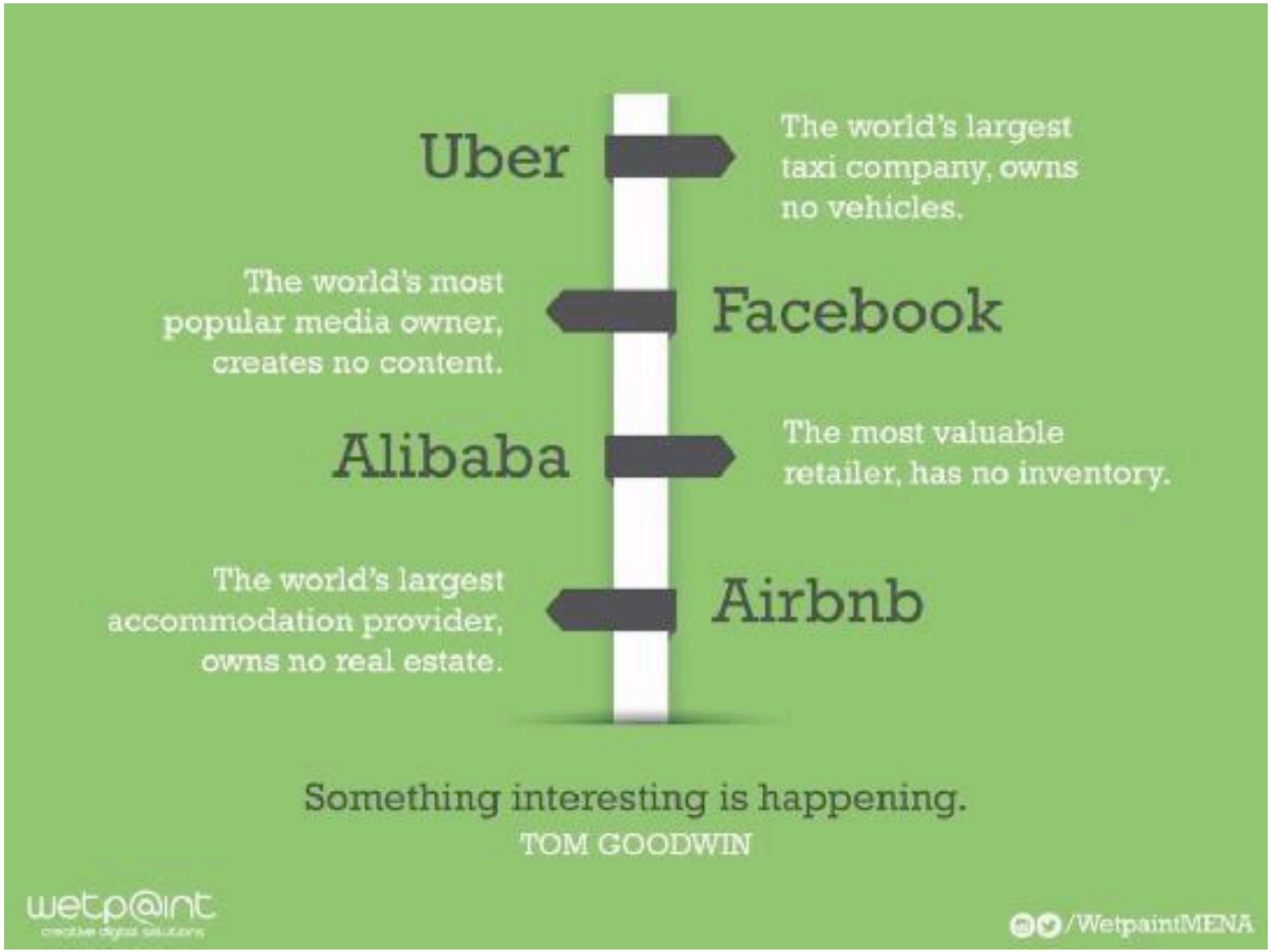
What to be careful of?

- Automation scope/feature creep
  - “Puppet can do anything!!”
- Not an orchestration/workflow language
- Traditional thinking vs “DevOps” thinking

  --> Robust first --> Speed first

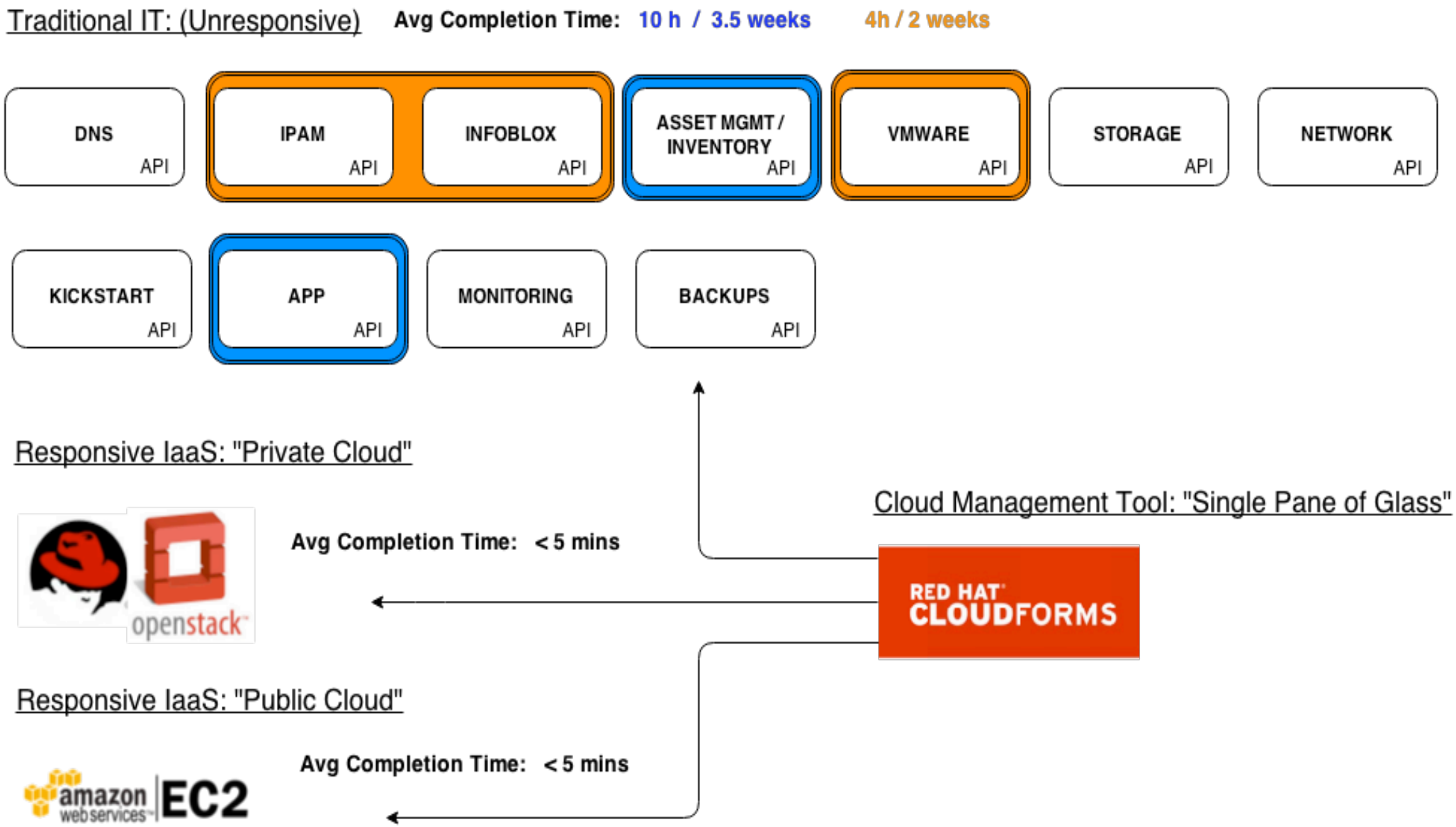
  --> Speed second --> Robust second

Unconventional questions from others…



Path to Private Hybrid Cloud

Server Provisioning Artifacts:



To Summarize:

- Automation and Orchestration are separate concepts with separate toolsets
- Courageous failures will provide tremendous value
- Unconventional thinking should be front of mind

Additional Resources

- Puppet Architecture - Craig Dunn, PuppetLabs Engineer
  - http://www.craigdunn.org/2012/05/239/
  - http://www.slideshare.net/PuppetLabs/roles-talk
- Puppet Training Classes
  - PuppetLabs Fundamentals
  - PuppetLabs Practitioner (Advanced)
- Guide to the Open Cloud
  - http://www.linuxfoundation.org/publications/linux-foundation/guide-to-the-open-cloud (PDF)
- RHEV vs VMware Pricing
  - http://www.redhat.com/en/files/resources/en-rhev-vs-vmware-vsphere-competitive-pricing-review-11717847.pdf

Thank You