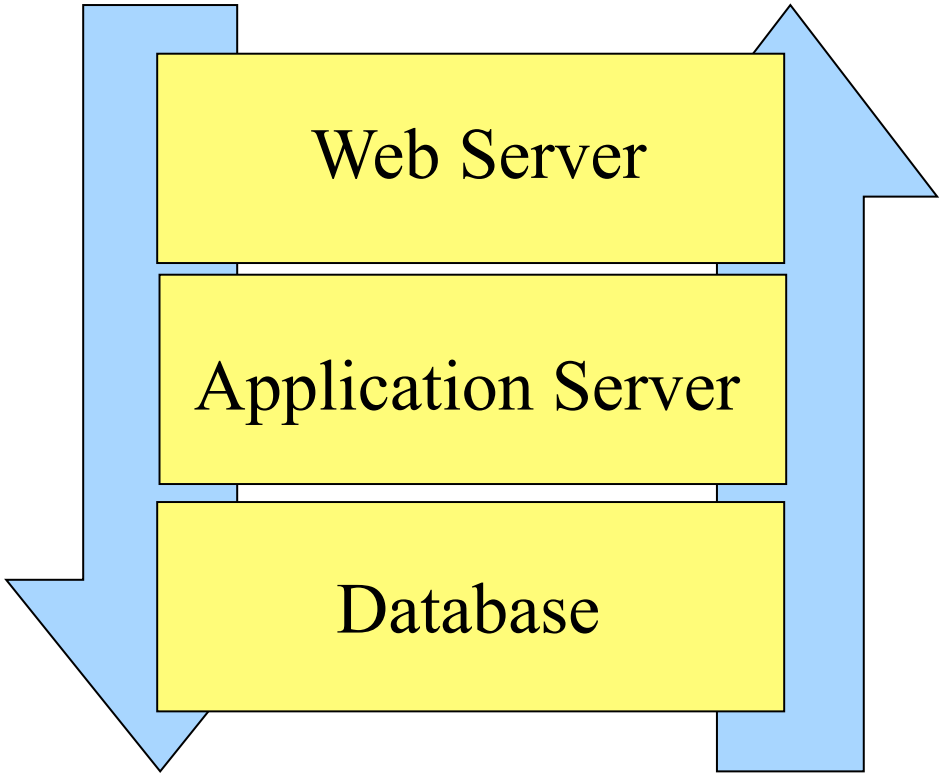
University of Victoria

Overcast: Eclipsing High Profile Open Source Initiatives ...seeding the clouds?

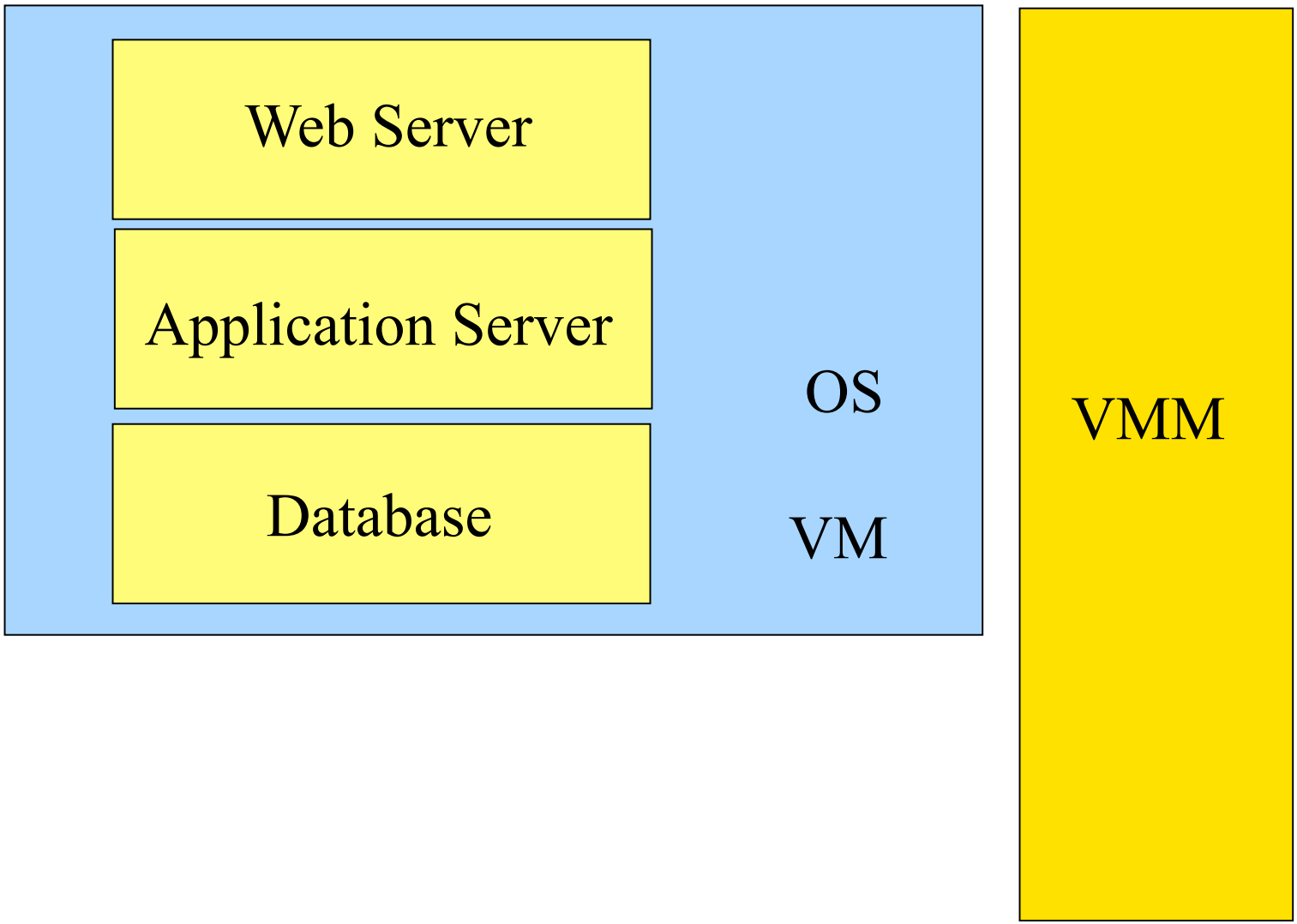
Chris Matthews, Stephen Neville,
Yvonne Coady, Jeff McAffer and Ian Bull

Classic

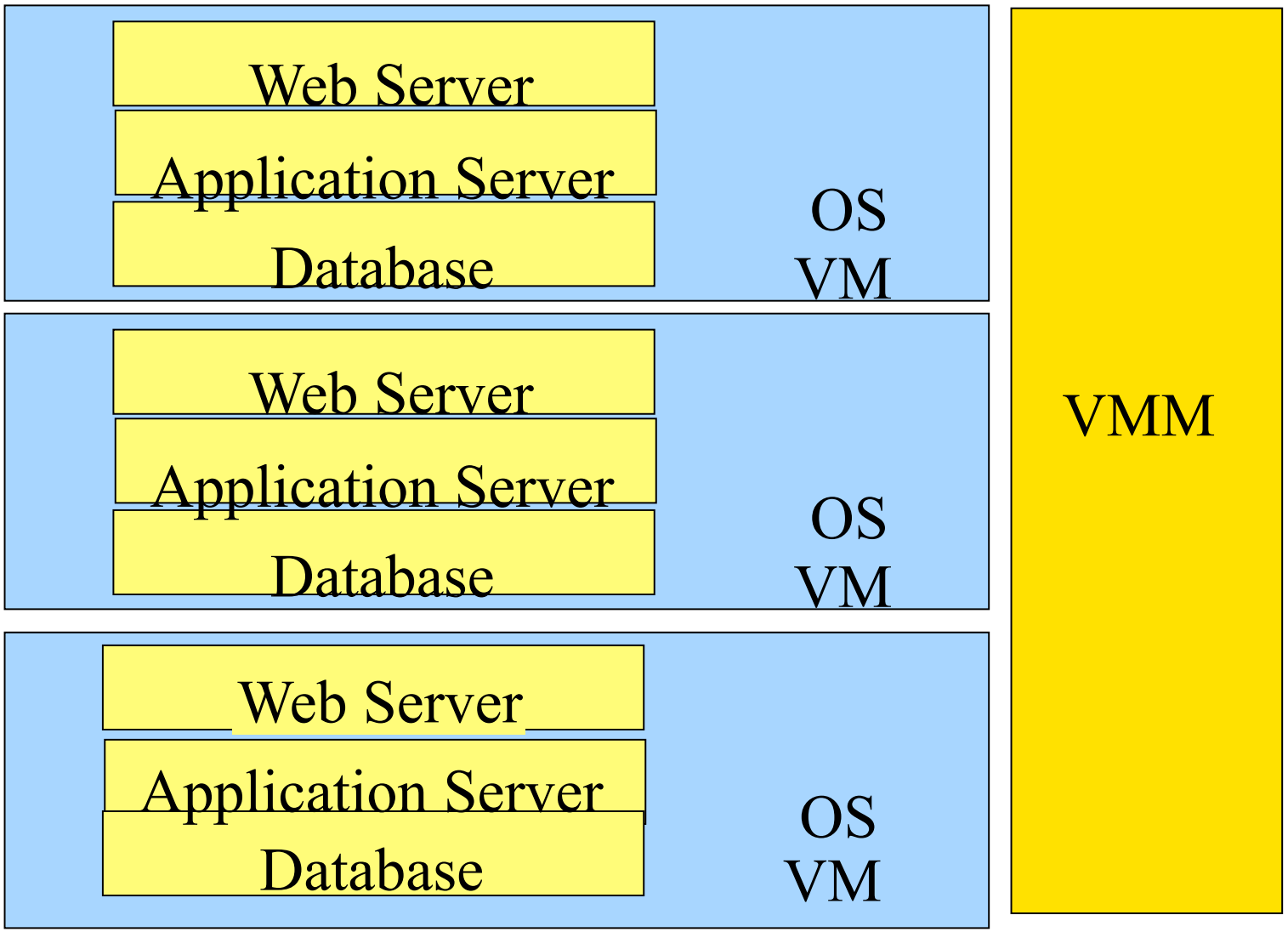
a simple web services stack...



Non-Virtualized

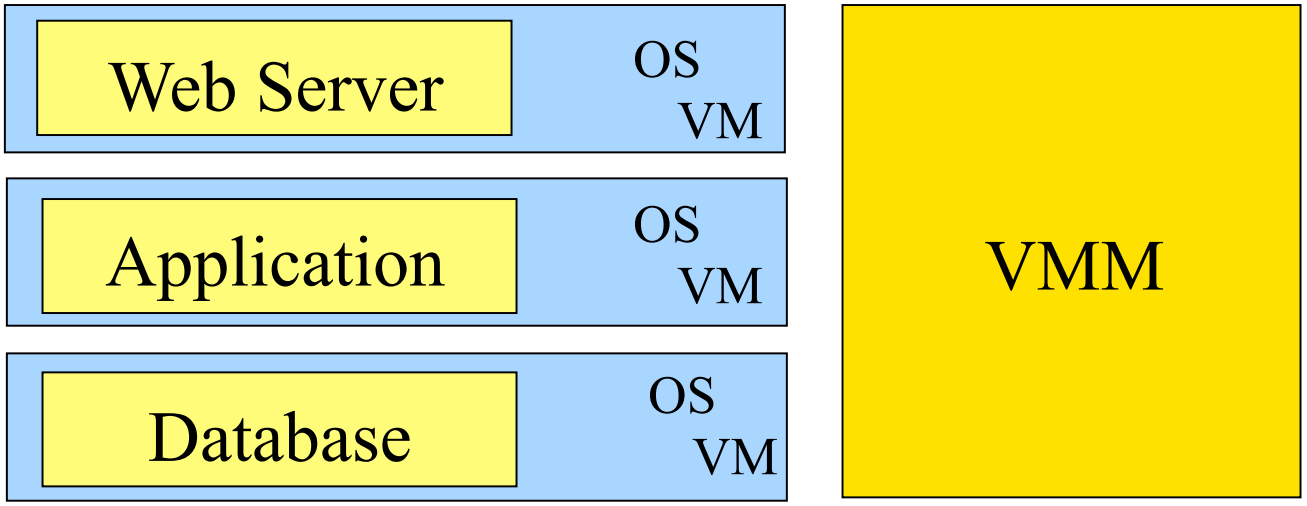


Non-Virtualized



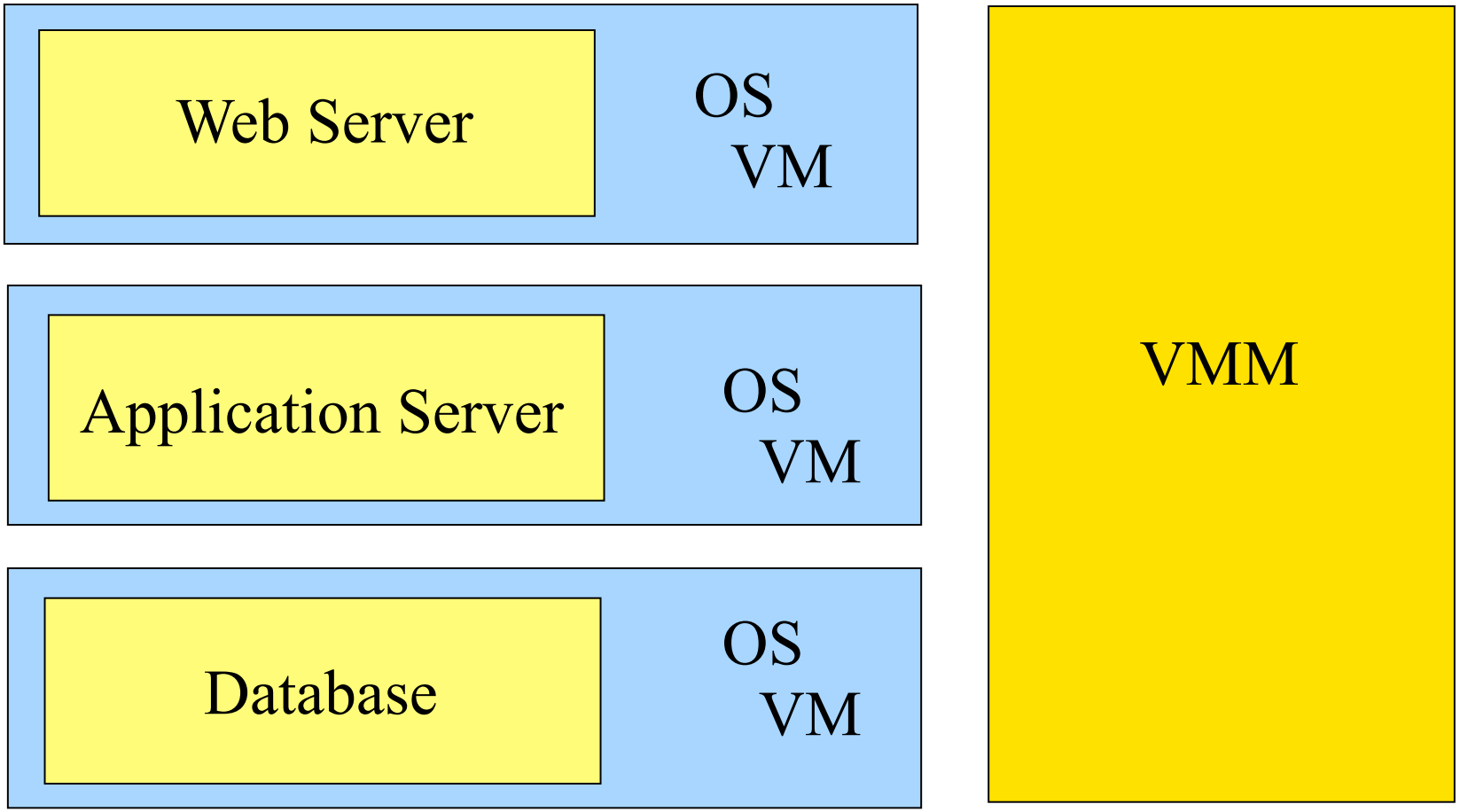
How Virtualization helps

- Functionality and Utilization
  - heterogeneity, hardware
- Security and Privacy
  - assertions, “black box”
- Reliability and Robustness
  - isolation, snap shoting
- Decomposition
  - migration, conceptual model



Decomposition

... More of a good thing?
Further decompose into domains?

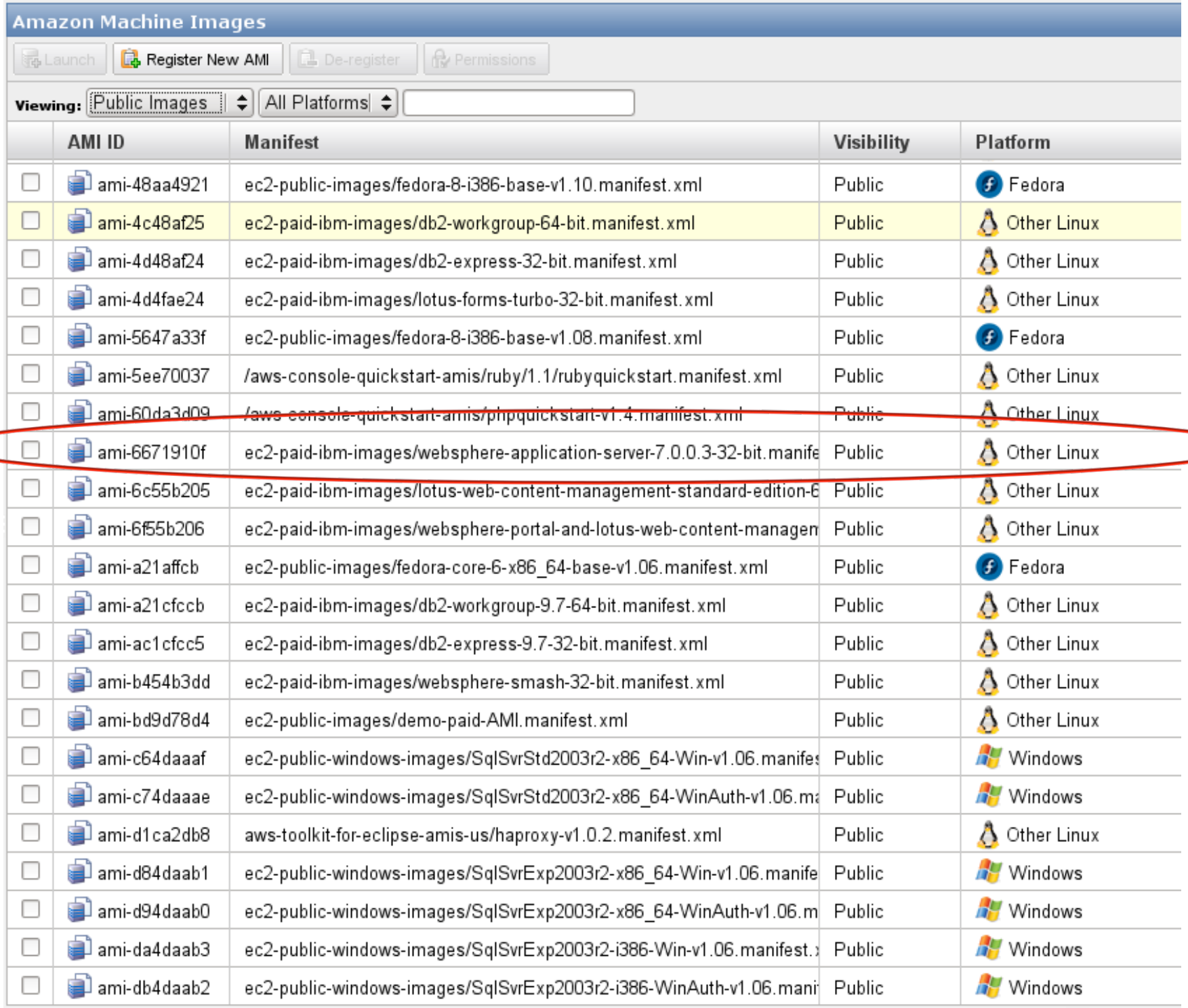


Road Map

- Today's virtual appliances and the cloud
  - Easybake appliances!
- Proposed approach
  - why is Chris trapped in a basement in BC?
- Tools that might help
  - CloudClipse = OSGi + P2 + Virtualization
  - Overcast
- How we can build better software for the cloud?
  - Software engineering challenge

Virtual Appliances

- Services prebuilt into VM images
- Building monolithic VM images is easy
- Amazon's AMI
- rBuilder from rPath
- OpenQRM
- Convenient interfaces for constructing!



Amazon Machine Images

Launch | Register New AMI | De-register | Permissions

Viewing: Public Images | All Platforms

	AMI ID	Manifest	Visibility	Platform
☐	ami-48aa4921	ec2-public-images/fedora-8-i386-base-v1.10.manifest.xml	Public	Fedora
☐	ami-4c48af25	ec2-paid-ibm-images/db2-workgroup-64-bit.manifest.xml	Public	Other Linux
☐	ami-4d48af24	ec2-paid-ibm-images/db2-express-32-bit.manifest.xml	Public	Other Linux
☐	ami-4d4fae24	ec2-paid-ibm-images/lotus-forms-turbo-32-bit.manifest.xml	Public	Other Linux
☐	ami-5647a33f	ec2-public-images/fedora-8-i386-base-v1.08.manifest.xml	Public	Fedora
☐	ami-5ee70037	/aws-console-quickstart-amis/ruby/1.1/rubyquickstart.manifest.xml	Public	Other Linux
☐	ami-60da3d09	/aws-console-quickstart-amis/phpquickstart-v1.4.manifest.xml	Public	Other Linux
☐	ami-6671910f	ec2-paid-ibm-images/websphere-application-server-7.0.0.3-32-bit.manife	Public	Other Linux
☐	ami-6c55b205	ec2-paid-ibm-images/lotus-web-content-management-standard-edition-6	Public	Other Linux
☐	ami-6f55b206	ec2-paid-ibm-images/websphere-portal-and-lotus-web-content-managen	Public	Other Linux
☐	ami-a21affcb	ec2-public-images/fedora-core-6-x86_64-base-v1.06.manifest.xml	Public	Fedora
☐	ami-a21cfccb	ec2-paid-ibm-images/db2-workgroup-9.7-64-bit.manifest.xml	Public	Other Linux
☐	ami-ac1cfcc5	ec2-paid-ibm-images/db2-express-9.7-32-bit.manifest.xml	Public	Other Linux
☐	ami-b454b3dd	ec2-paid-ibm-images/websphere-smash-32-bit.manifest.xml	Public	Other Linux
☐	ami-bd9d78d4	ec2-public-images/demo-paid-AMI.manifest.xml	Public	Other Linux
☐	ami-c64daaaf	ec2-public-windows-images/SqlSvrStd2003r2-x86_64-Win-v1.06.manifes	Public	Windows
☐	ami-c74daaae	ec2-public-windows-images/SqlSvrStd2003r2-x86_64-WinAuth-v1.06.ma	Public	Windows
☐	ami-d1ca2db8	aws-toolkit-for-eclipse-amis-us/haproxy-v1.0.2.manifest.xml	Public	Other Linux
☐	ami-d84daab1	ec2-public-windows-images/SqlSvrExp2003r2-x86_64-Win-v1.06.manife	Public	Windows
☐	ami-d94daab0	ec2-public-windows-images/SqlSvrExp2003r2-x86_64-WinAuth-v1.06.m	Public	Windows
☐	ami-da4daab3	ec2-public-windows-images/SqlSvrExp2003r2-i386-Win-v1.06.manifest.x	Public	Windows
☐	ami-db4daab2	ec2-public-windows-images/SqlSvrExp2003r2-i386-WinAuth-v1.06.mani	Public	Windows

rBuilder Online BETA

rPath Management Console

Provide feedback Old UI

Resources

Appliances
Ready-to-deploy applications

Platforms
Foundations for appliances

Repositories
Packages for a variety of uses

Notices

rBuilder Updated: Interfa...

rBuilder Online maintenance is complete, and the following features and bug fixes are now available.

- rBuilder now presents two viewing options for appliance search results: a "Brief" view

rBuilder Online Scheduled...

rBuilder Updated: Deep Li...

Appliances

Brief | Detailed

Amiga 100 Software Platform

Another Custom Linux
Just having a play to check this thing out at this stage

Apache Appliance
A small Linux build designed only to support httpd. The image contains the minimal group-core from rPath Linux and an Apache web server (httpd) that s

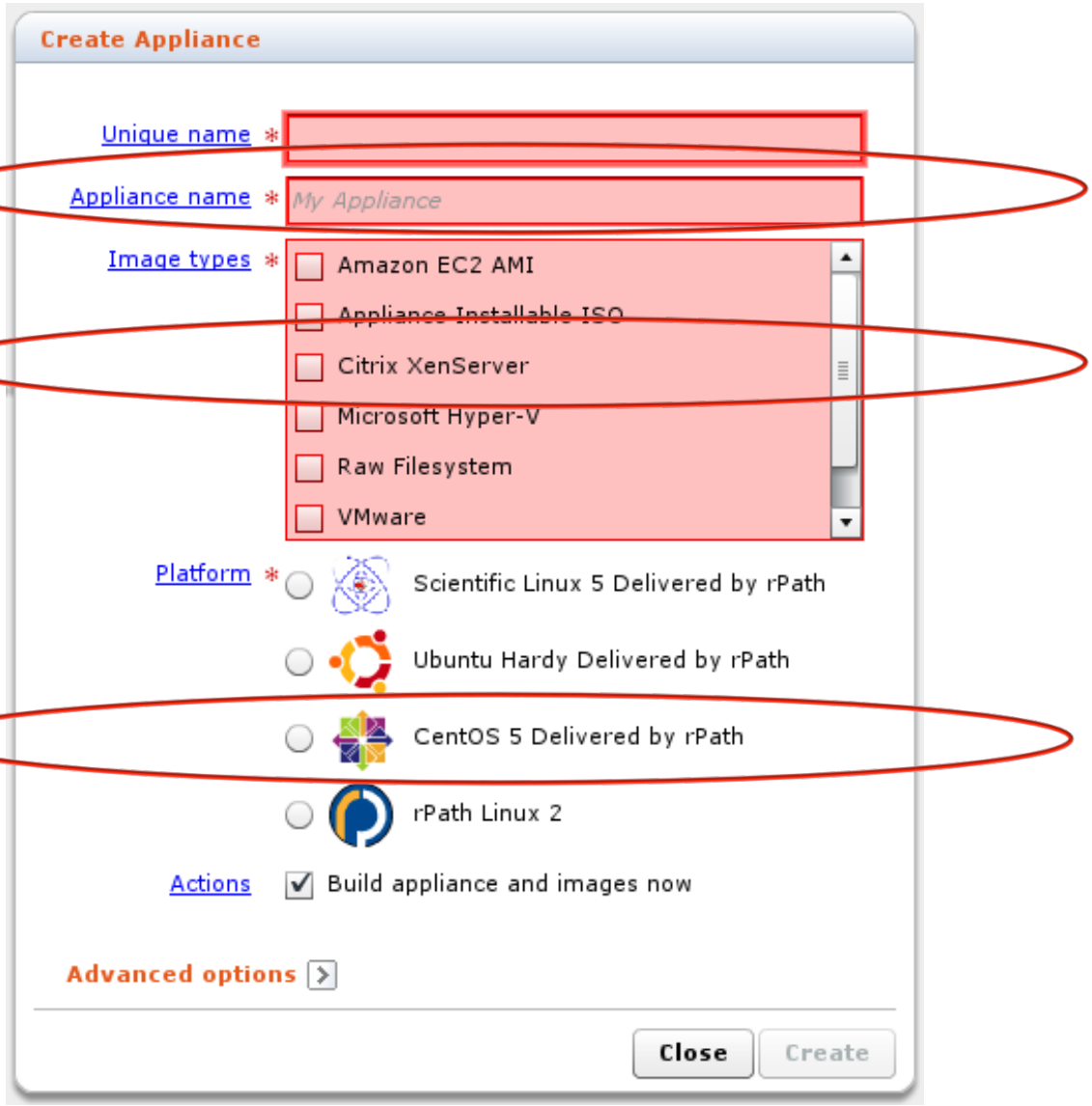
It also include openssh client and server, vim and slocate for convenience.

Asterisk 2 Billing Live CD

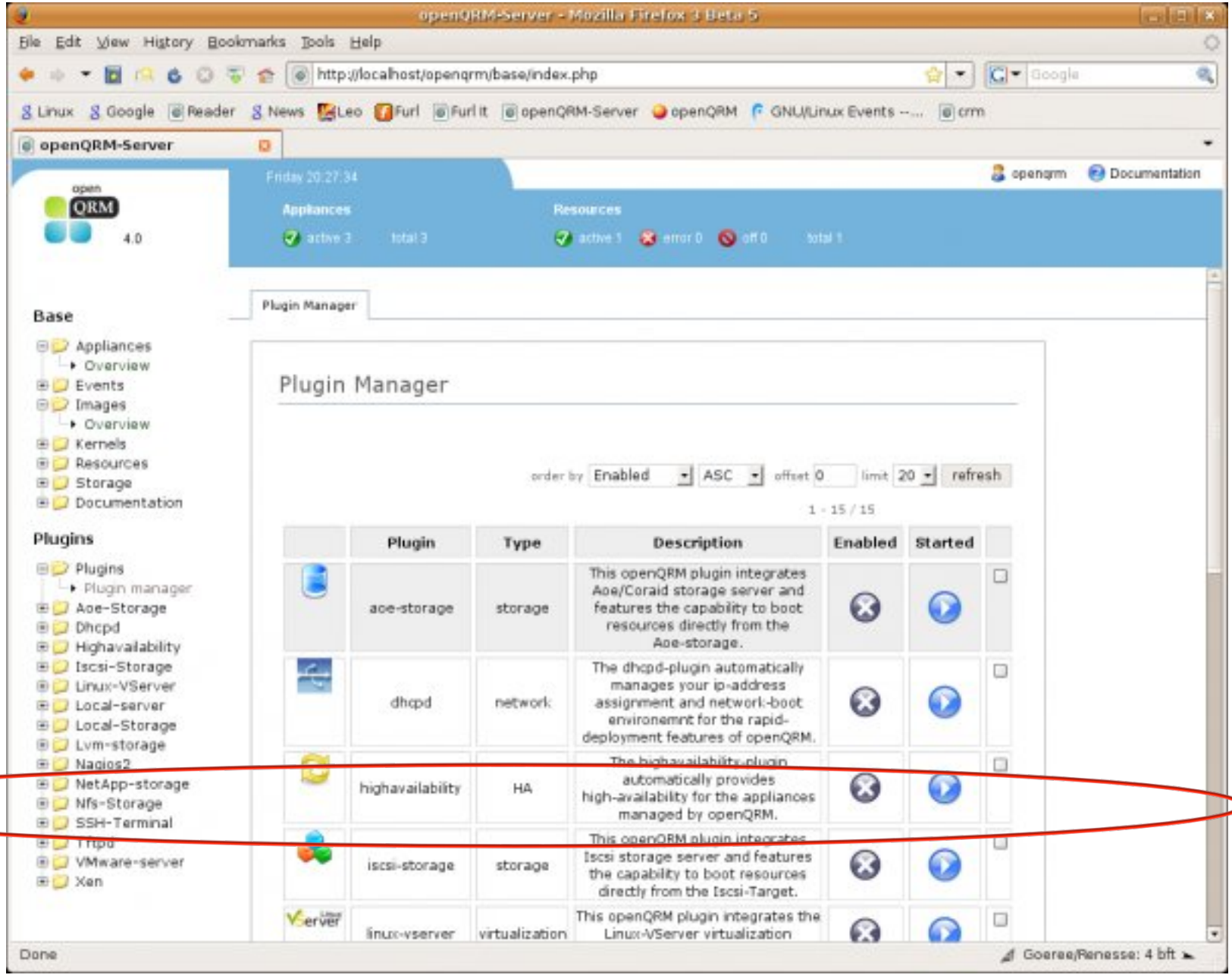
Asterisk @ Home
Asterisk @ Home

Asterisk+Apache + MySQL+PHP
Asterisk+Apache + MySQL+PHP

Create an Appliance from your Application

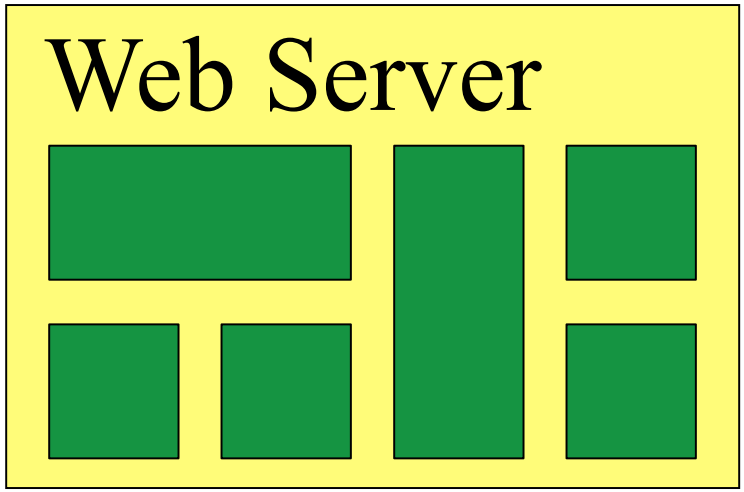


“Plugins” in OpenQRM



Proposed Approach

- What if we tried to further decompose the system?



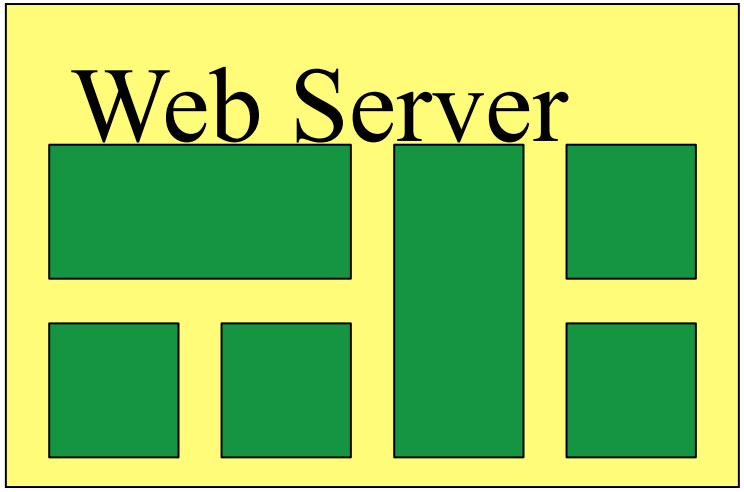
- **Virtual Co-processors**
  - firewall
  - encryption
  - random number generator
  - TPMs
- **Services**
  - logging
  - time stamp
  - inspection monitor

Principles for decomposition...

C[4]: The Cs system decompostion

Componentization
Composition
Communication
Control

To enable fine-grained components in a system.



Componentization

- Disaggregate the elements of the system into components
  - Modern day modularity?
  - What will change in the system?

- These components benefit from the properties of virtualization

- Must trade off with the intrinsic costs
  - How big are they?
  - How are they packaged?

Composition

- Ideally we want dynamic programmatic composition of domains

- We need mechanisms for
  - referencing other components
  - exploring component's metadata and interfaces
  - creating, updating, distributing them
  - destroying them
  - customization

  ...BUT who does the composition?
  ...AND how is the composition done?

Communication

- Communication should be as easy as IPC or a function call

- The system needs to provide
  - some plumbing between domains
  - naming and registration
  - interfaces to describe services and communication
  - defined communication semantics
  - security services

- Need secure, reliable, cross domain communication that is easy to use
  - what are the communication semantics?

Control

- The system should be able to define policies and enforce them with mechanisms
  - controlling life-cycle
  - monitoring resource usage

what mechanisms are needed to correctly control a component?

where are these mechanisms controlled from?

where are the policies of a component defined?

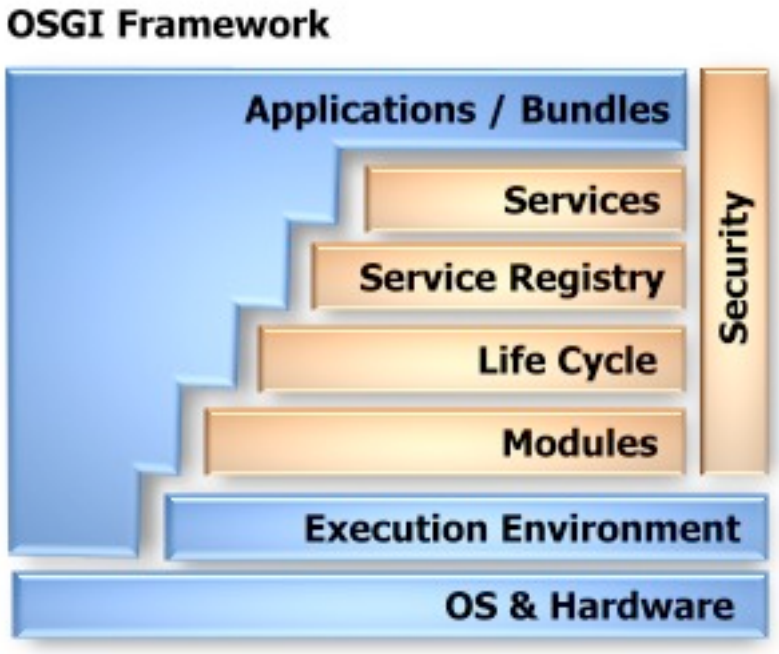
So Why is Chris Locked in a Basement?

- As the decomposition becomes more fine-grained the attendant costs of the decomposition increase in two ways
  - Scalability
  - Sustainability

- High burden of composition
  - on the system resources...
  - on the programmer...

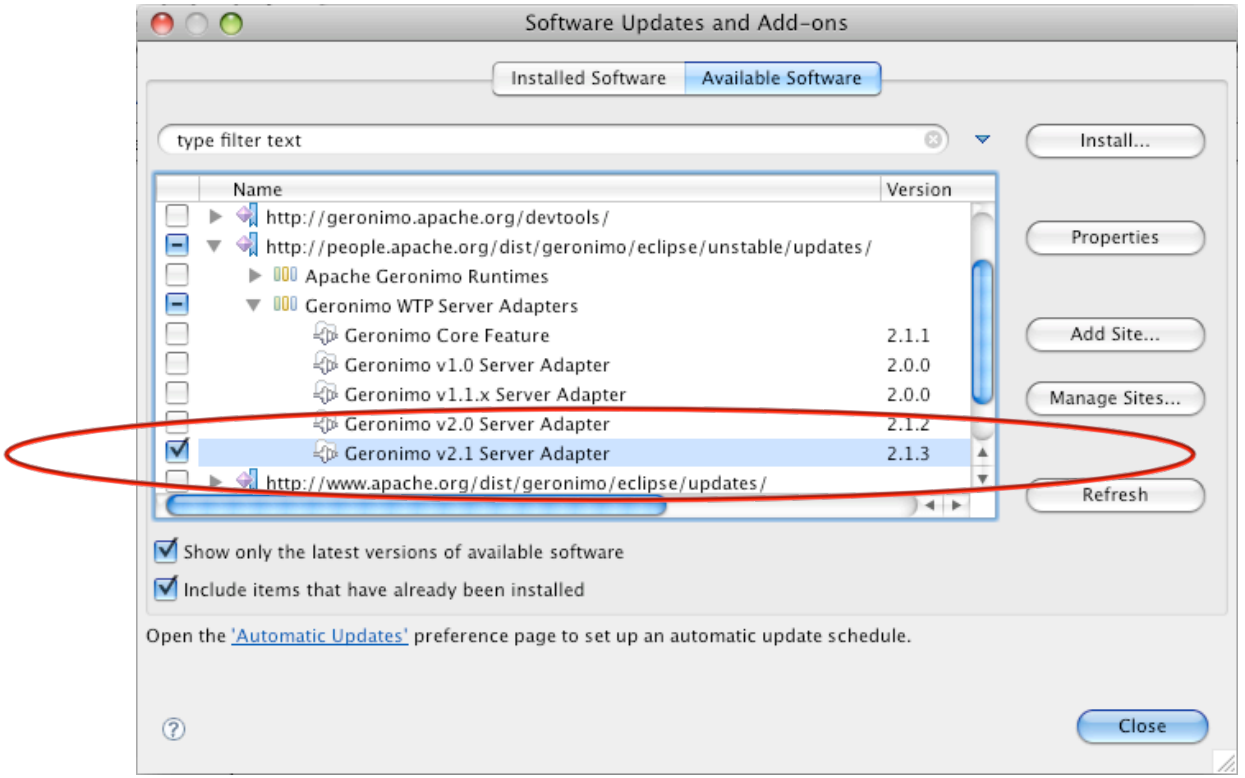
Leveraging Existing Technology: OSGi

- Popular Java component model
- Provides services, naming, versioning and lifecycle management
- Dynamic service model maps to distributed systems (R-OSGi)
- Eclipse Foundation provides reference implementation named Equinox
- Equinox powers the Eclipse IDE



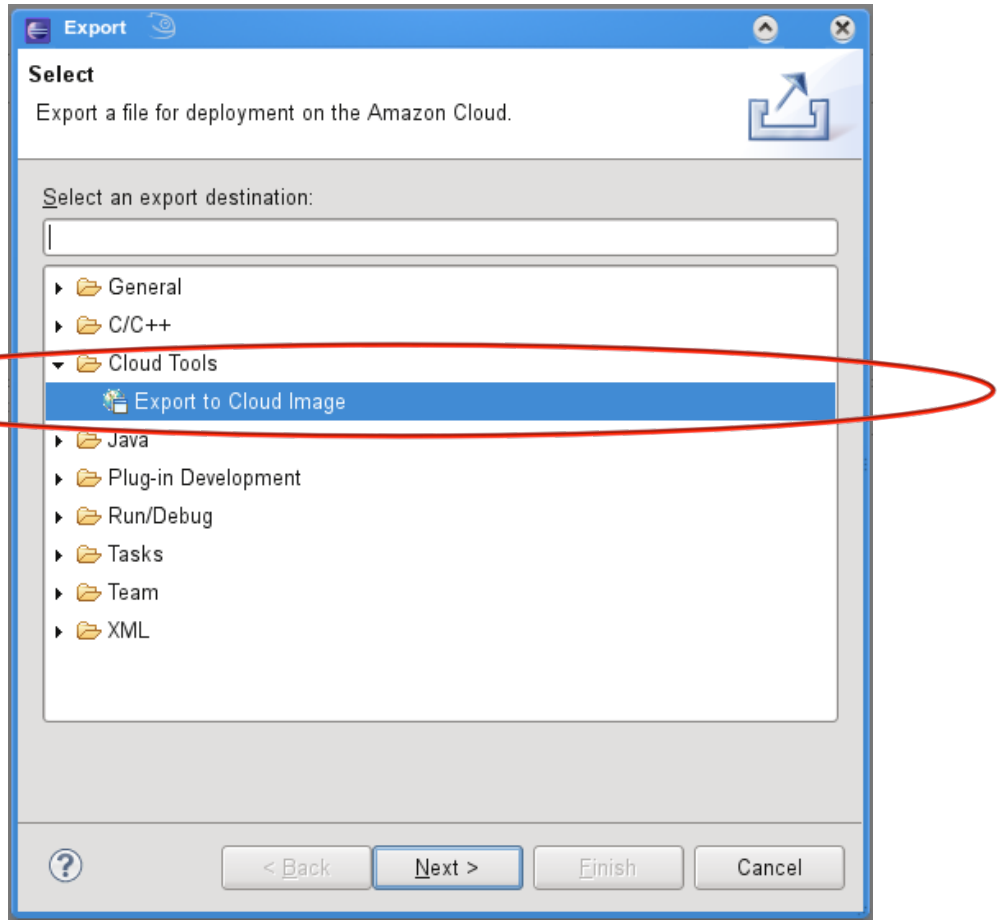
P2: a provisioning platform

- A provisioning system for OSGi applications
- In some ways similar to a package management system
- Used in the Eclipse IDE update mechanism



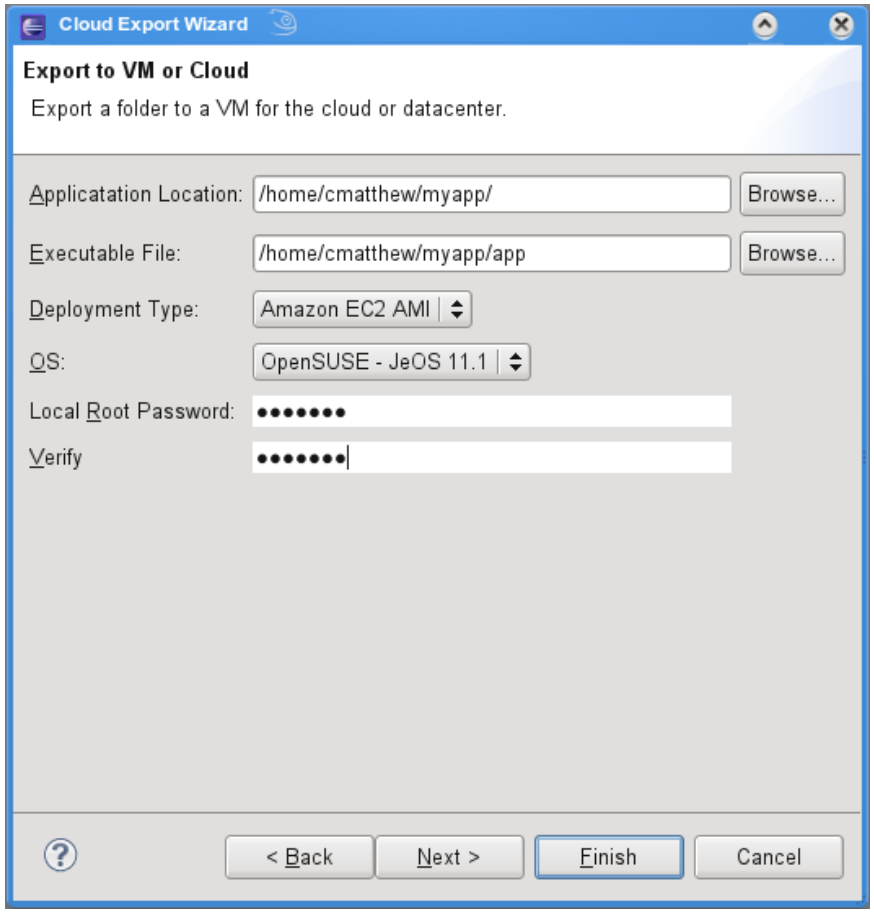
CloudClipse

- Google Summer of Code
  - Eclipse plugin
- Combines the power of P2 and OSGi with virtualization
- Extends the P2 engine with the tools to build VM images



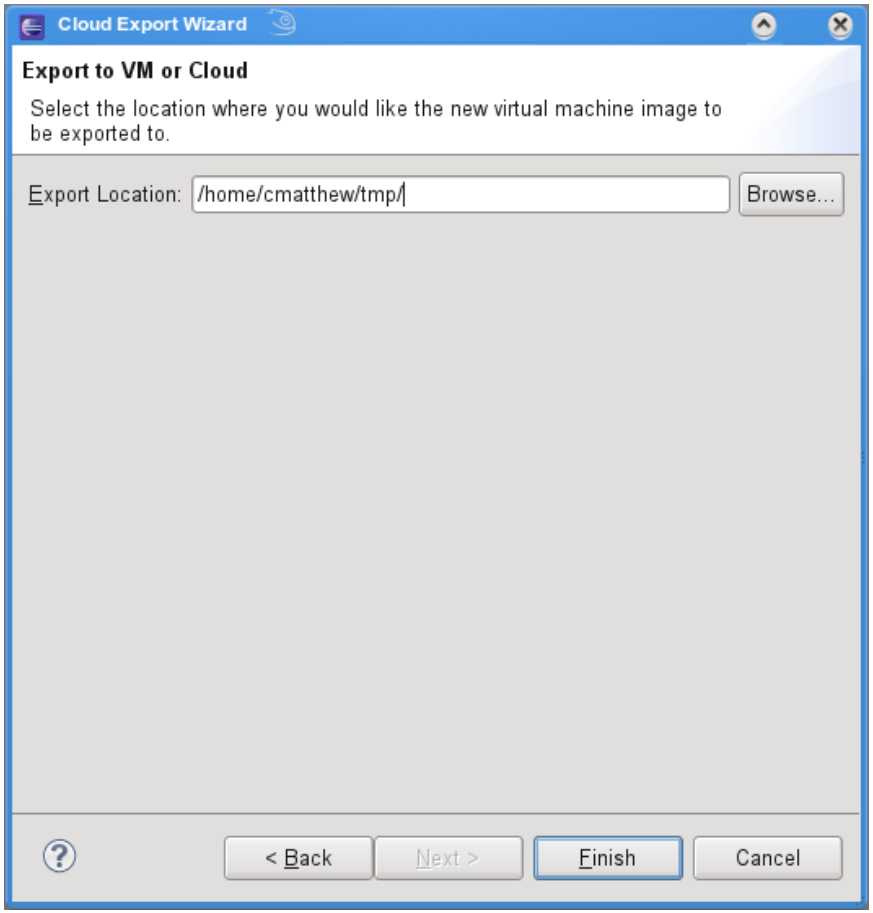
CloudClipse

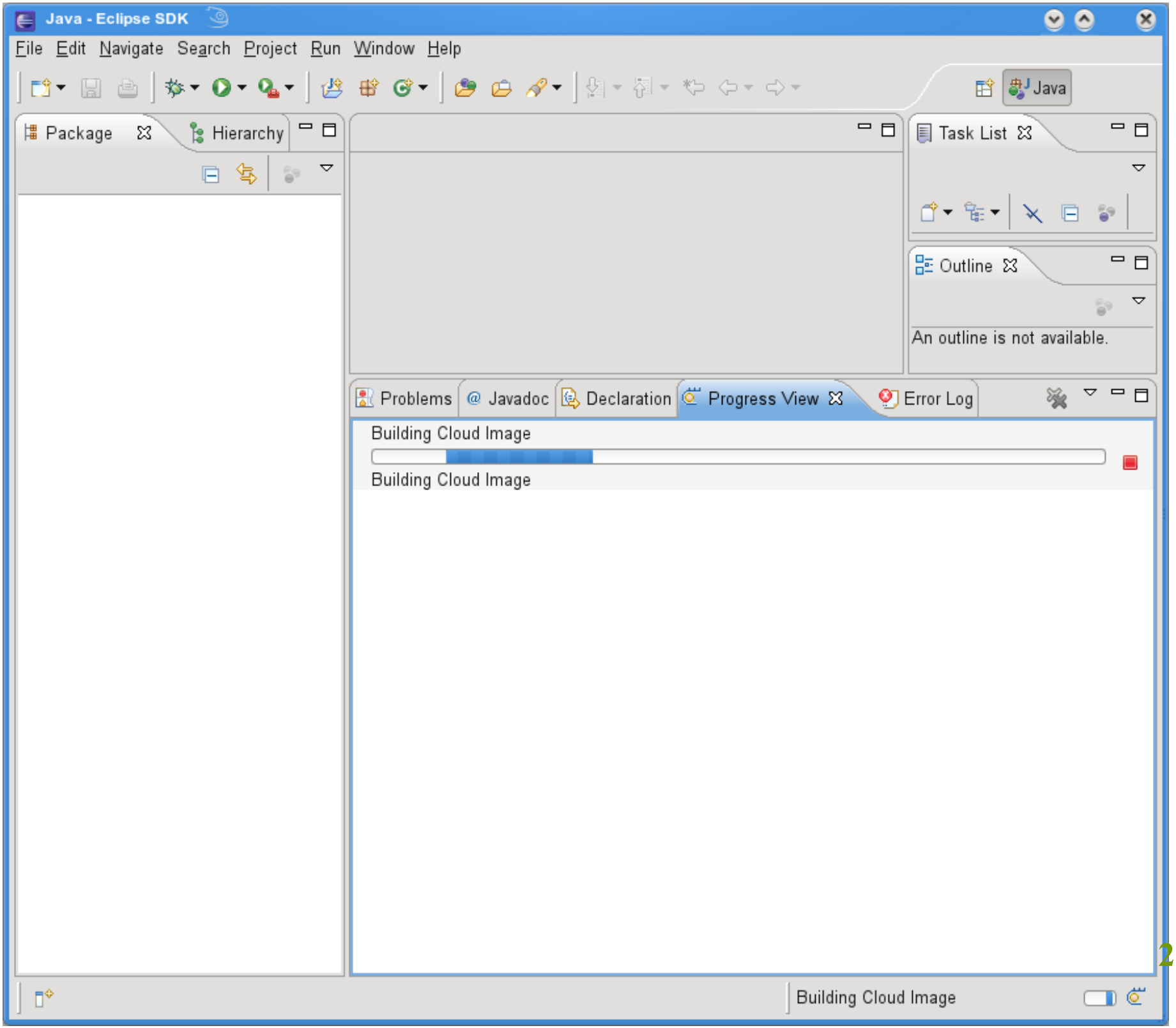
- CloudClipse extends the P2 installer to:
  - Mount images
  - Install RPMs
  - Copy new files in
  - Run commands in the image with chroot
  - Compact images



CloudClipse

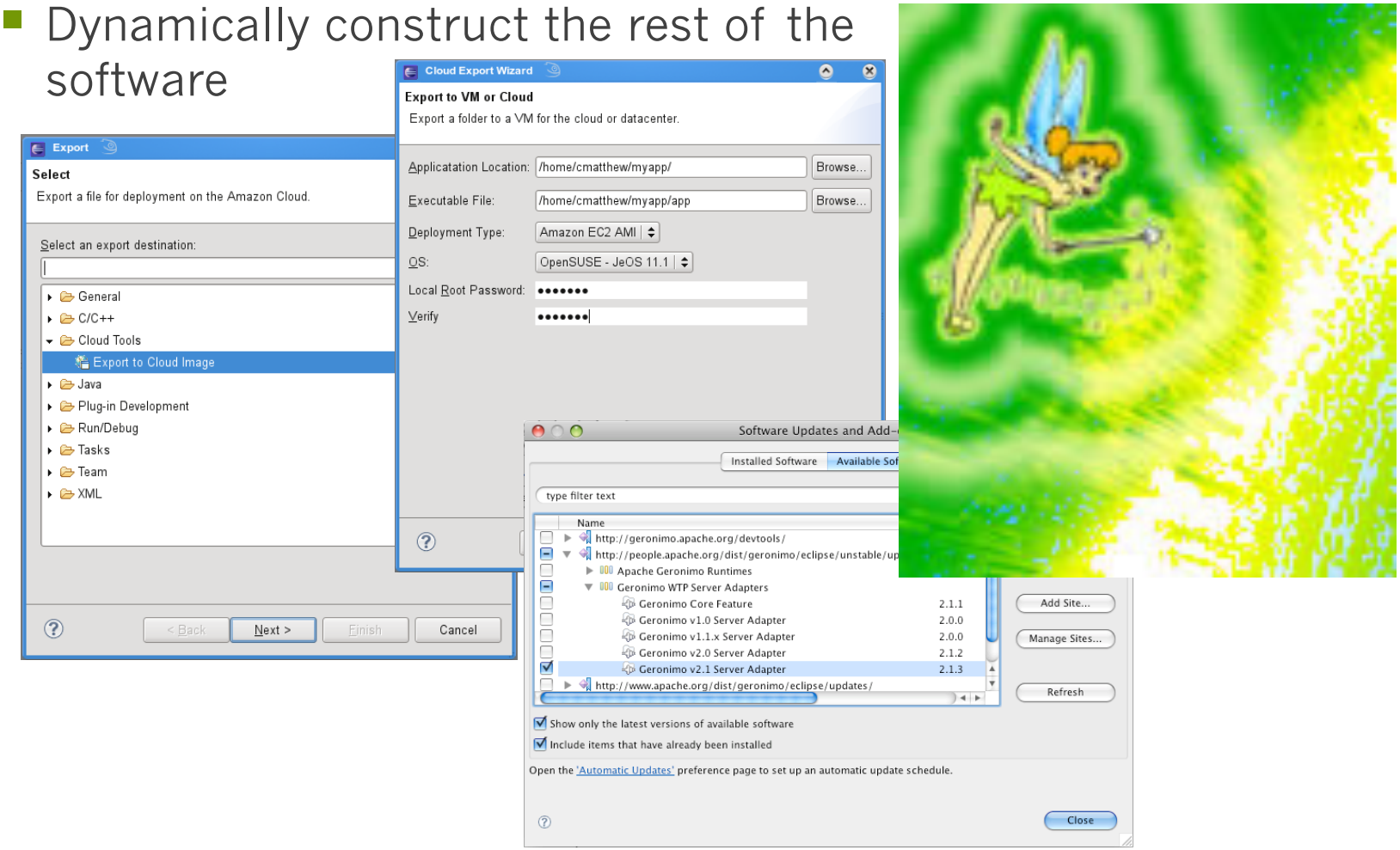
- Basic Operation:
  - P2 finds and downloads a base image (to save time)
  - Image is mounted
  - RPMs are installed
  - Config files are setup
  - Passwords are set, ssh keys are installed
  - Your application is installed




Java - Eclipse SDK
File Edit Navigate Search Project Run Window Help
Java
Package
Hierarchy
Task List
Outline
An outline is not available.
Problems
Javadoc
Declaration
Progress View
Error Log
Building Cloud Image
Building Cloud Image
Building Cloud Image

Overcast: The Vision!

- Boot strapping!
- Create an image with P2 within it
- Dynamically construct the rest of the software



YES WE CAN Leverage Current Component Systems... we think!

- OSGi provides an interesting component model for MacroComponents and more generally Cloud Service Compositions
- R-OSGi shows dynamic services could work as distributed services
- Preexisting tool support for OSGi

- Still lacking Communication and Control systems

- ....This could allow us to build and deploy better software for the cloud!

Summary and Conclusions

- Want to tap into the software engineering benefits of components
  - strongly isolated
  - redundancy is easily inserted into the system
  - hot standbys are cheaper
- Overall a more robust secure and scalable software system of heterogeneous components
- Breaks the dependence on large software stacks
- Match conceptual models!
- Challenges:
  - Adoption
  - Privacy
  - Overheads

Chris: one day he'll have a REAL Parrot!

Questions

www.christophermatthews.ca