



Eclipse Virgo Update

Glyn Normington

`gnormington@vmware.com`

Agenda

- Why use Virgo?
- What does it give me?
- Where has it gotten to?
- How can I get started?

Why use Virgo?

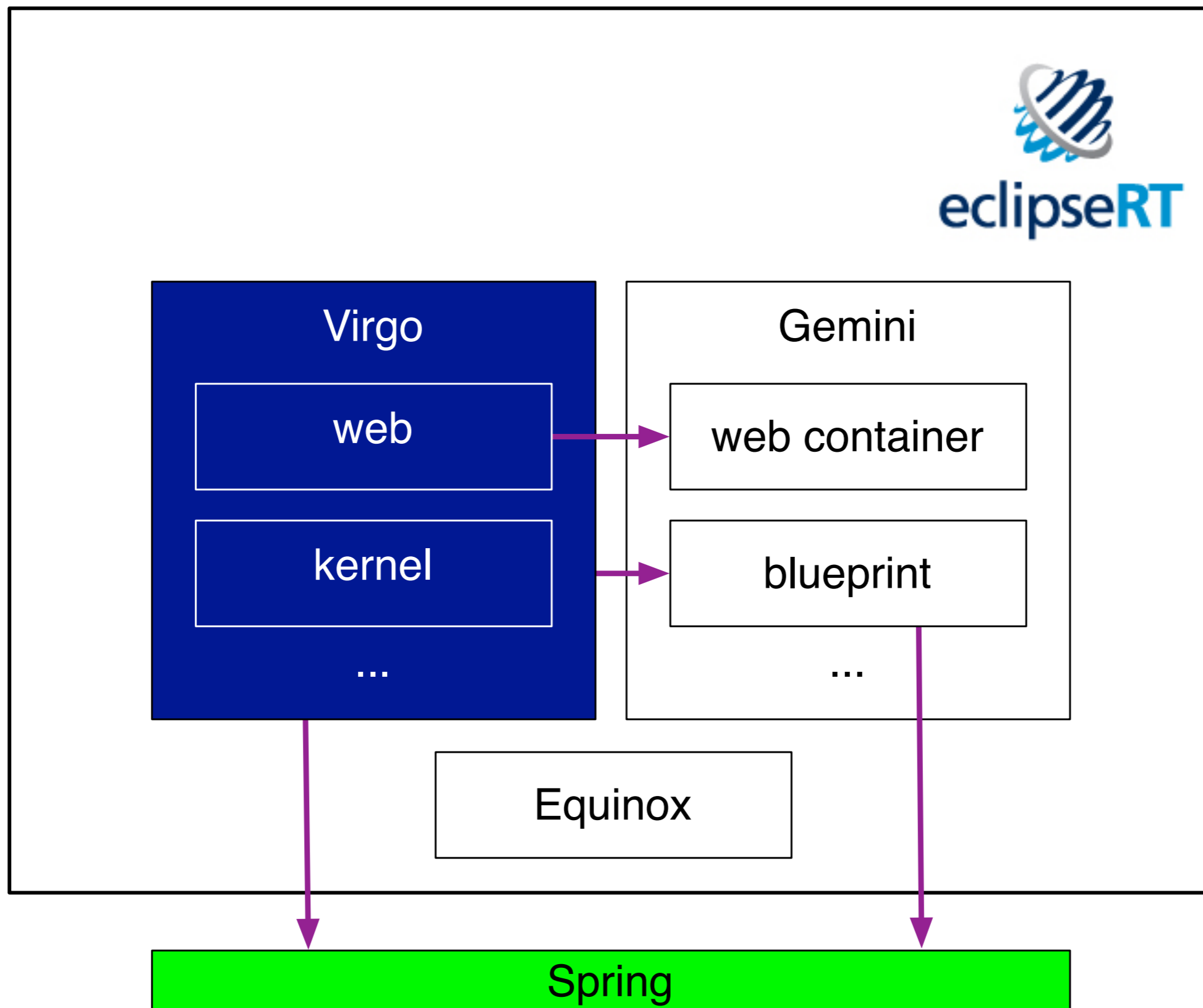
Why use Virgo?

- Based on OSGi standards
 - Core v4.2
 - Compendium v4.2
 - Enterprise v4.2
- Built from OSGi bundles
- Supports OSGi applications
 - “Vanilla” bundles
 - Web bundles
- Tracking future OSGi standards

Why use Virgo?

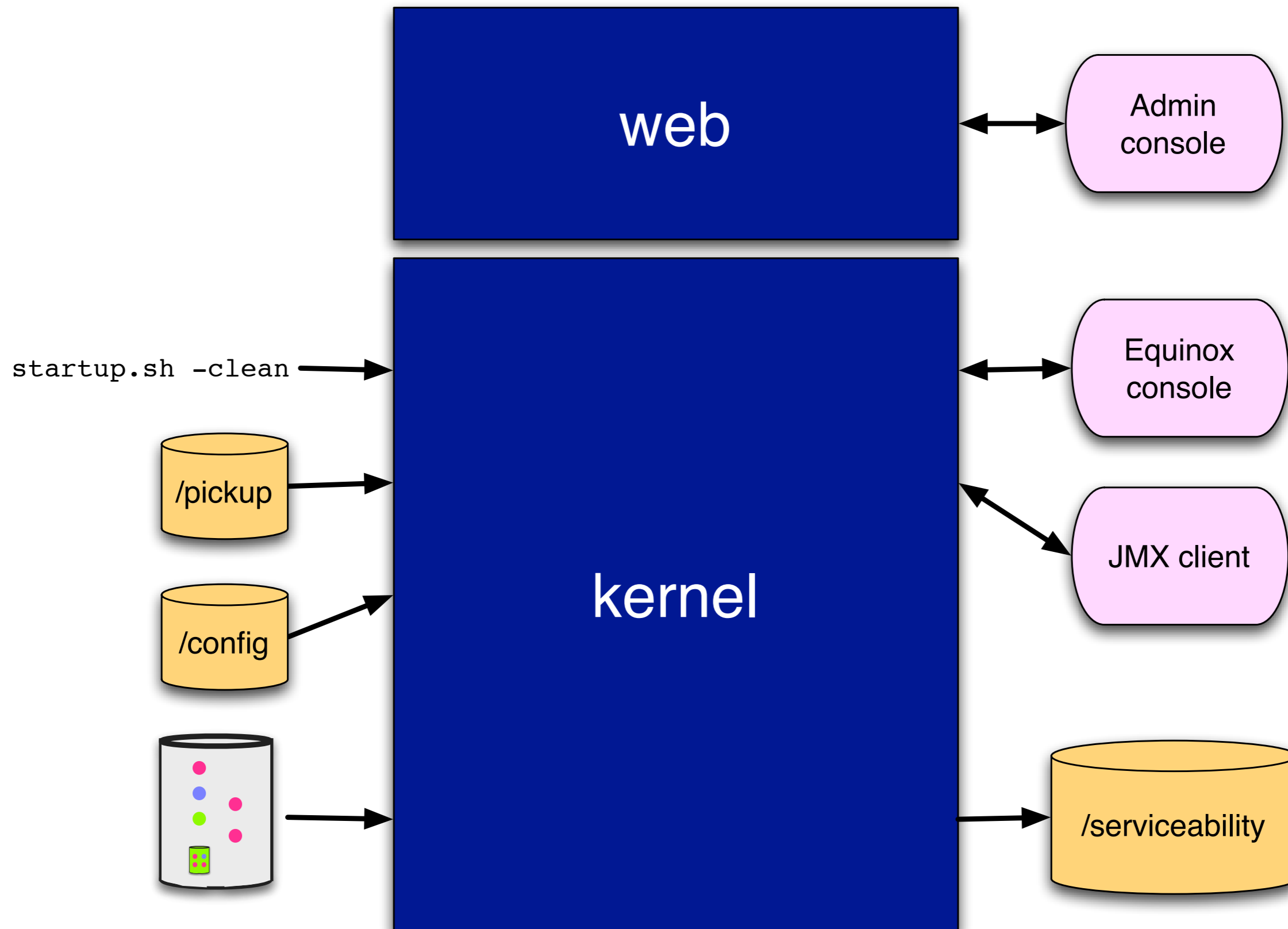
- Grouping *artefacts* into *plans*
- Auto-provisioning dependencies
- Use existing libraries
- Management: web, console, JMX
- Advanced diagnostics
- Kernel and web server
- Extensible deployment pipeline

Virgo and Gemini



What does Virgo give me?

Virgo Runtime



Virgo non-runtime

- SpringSource Tool Suite support
- Bundlor
- Samples
- Documentation
- Test suite
- Blogs, presentations, webinar, ...

Where has Virgo gotten to?

Virgo runtime

- All code now in Eclipse GIT
- Milestones shipping regularly
- IP in the last stages
- All repositories can be checked out and built (except SVTs - to follow)
- Hudson CI server

First release

- 2.1.0 release next month
- Migration from dm Server 2.0.x
- Major features:
 - 5x startup performance improvement
 - Tomcat, Logback, Equinox upgrades
 - Equinox console integration

Virgo tooling

- SpringSource Tool Suite
 - Update site
 - Donation to follow
- Fast deploy/update/debug cycle

Community

- 3 committers
- 5 further code contributors
- VMware, SAP, others

How can I get started?

Getting started

- Pre-req: Java 6
- Download and unzip
- bin/startup.sh or bin\startup.bat
- drop WAR files or bundles into pickup

...then what?

- Read the documentation
 - User Guide
 - Programmer Guide
 - Tutorial “Creating an App with Virgo”
- Split out dependencies
- Join the community

Further Information

- <http://www.eclipse.org/virgo>



The screenshot shows the Virgo project website. On the left is a navigation menu with links for 'Download', 'Documentation', 'Support', 'Getting Involved', and 'Background'. The main content area features the Virgo logo (a stylized globe) and the text 'VIRGO from eclipseRT'. Below the logo, there are two paragraphs of introductory text. The first paragraph describes the Virgo Web Server as a module-based Java application server. The second paragraph describes the Virgo kernel as a core concept that supports various server types. Below this is a list of features, including an admin console, Equinox console extension, plans, provisioning, web container, user region, Spring 3 support, hot deployment, and logging. On the right side of the page, there are sections for 'Quick Links' (listing Virgo Wiki, Forum, Mailing List, Summary, Bugzilla, and CI Server), 'Blogs' (listing authors like Chris Frost and Glyn Nomington), 'Presentations' (listing a webinar from August 2010), 'Virgo Powered' (listing the TINOS research project), and 'Incubation' (featuring an Eclipse incubation logo). At the bottom right, there is a social media section titled 'What's being said about... #VirgoRT' with a tweet from cgfrost about updated download pages.

Virgo Home »
About Virgo

- Download
- Documentation
- Support
- Getting Involved
- Background



The Virgo Web Server from EclipseRT is a completely module-based Java application server that is designed to run enterprise Java applications and Spring-powered applications with a high degree of flexibility and reliability. It offers a simple yet comprehensive platform to develop, deploy, and service enterprise Java applications.

The Virgo kernel supports the core concepts of Virgo and is not biased towards the web server, thus enabling other types of server to be created. The kernel can also be used stand-alone as a rich OSGi application platform. A server runtime can easily be constructed by deploying suitable bundles on top of the kernel.

Virgo features:

- admin console - deploy and manage artifacts, examine diagnostic dumps, and examine bundle wiring, either in the live system or from a resolution failure
- Equinox console extension - manage Virgo and deployed artifacts
- plans - define the artifacts that comprise an application, optionally making the application atomic to tie the artifact lifecycles together, and scoped to isolate the application from other applications
- provisioning - automatically supply the dependencies of an application including bundles, plans, plan archives (PARs), and configurations, from both local and remote repositories
- web container - supports vanilla WAR files, with all their dependencies in WEB-INF/lib, and Web Application Bundles, that import their dependencies via OSGi manifest metadata, via the embedded Tomcat-based reference implementation of the OSGi Web Container specification, configured using the standard Tomcat `server.xml`
- user region - isolates the kernel from user-installed applications and enables admins to focus on application artifacts and their dependencies without seeing those of the kernel as well
- Spring 3 - Virgo packages Spring 3.0 final, but can easily be configured to use a different version of Spring
- hot deployment - deploy artifacts to Virgo by copying them into the `pickup` directory, either in archive or exploded form, as an alternative to deploying via the admin console
- logging - via embedded LogBack, configured in `config/serviceability.xml`, with a rich set of appenders available out of the box

Quick Links

- Virgo Wiki
- Virgo Forum
- Developer Mailing List
- Project Summary
- Virgo Bugzilla
- Hudson CI Server

Blogs

- Chris Frost
- Glyn Nomington
- Michael Coté
- Nitin Gupta
- Tommy McGuire

Presentations

- Virgo Webinar, 31 August 2010

Virgo Powered

- TINOS research project

Incubation



What's being said about...
#VirgoRT

 cgfrost Updated the #virgort download page to have previous shippings and release notes for each one. #osgi <http://bit.ly/cuKwOI>