



GridWeaver

Large-Scale, Adaptive Fabric Configuration for Grid Computing

Peter Toft
HP Labs, Bristol

June 2003 (v1.02)
Localised for UK English

The GridWeaver Project

- A collaboration between
 - Serrano Project, HP Labs Bristol (UK)
 - School of Informatics, University of Edinburgh
 - EPCC, University of Edinburgh
- Combining research interests and technologies from HP Labs (SmartFrog) and Edinburgh (LCFG)
- Funded by the UK e-Science Grid Core Programme, and by HP
 - A 1 year project, to July '03
 - Research-oriented, not building production systems



The Challenge

- An effective Grid assumes the existence of correctly operating, large-scale fabrics
 - From base OS on a single fabric element, up to complex, coupled, distributed services
- Every aspect of the fabric must be correctly configured
 - Scale
 - Diversity
 - Complexity
 - Dynamism
- Challenging problems:
 - Scale
 - Diversity
 - Complexity
 - Dynamism
- Aim to:
 - Describe whole-fabric configuration
 - Deliver a correctly-configured fabric
 - With automatic adaptive behaviour



Research Interests

- Expressing system configurations
 - Models and languages for representing configurations of resources
- Using and manipulating system configurations
 - Tools to assist in creating and manipulating correct configurations
- Deploying system configurations
 - Turning the representation of your desired system into a realised, running system
 - Keeping the configuration correct over time
- Creating adaptive system behaviour
 - A framework for automatic reconfiguration to accommodate changes (including failures)



Technologies: LCFG and SmartFrog

- A common philosophy ...
 - Language-based approaches for expressing whole-system configuration
 - Frameworks and extensible component sets for realising system configurations
 - “Asymptotic” configuration to deal with scale
- ... with complementary foci
 - LCFG focuses on configuring and managing individual nodes in a fabric
 - SmartFrog focuses on configuring and orchestrating distributed applications running across nodes

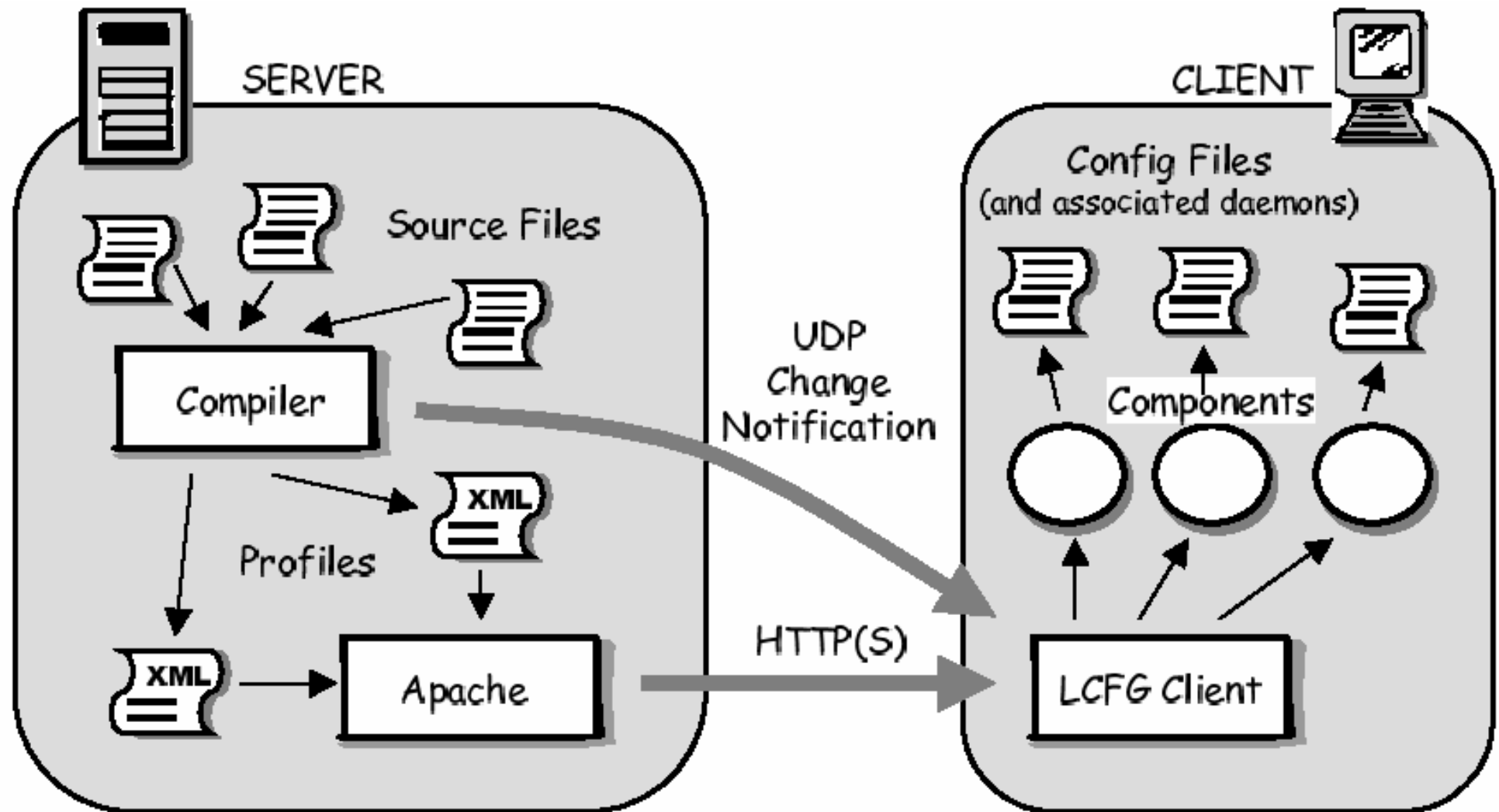


Overview of LCFG

LCFG: “Local Configuration System”

- Declarative, prototype-based language for defining fabric configurations
 - Configuration “aspects” are combined to create a complete configuration profile for each node
- Components for configuring and managing most aspects of node configuration
 - E.g.: configuring, starting, and stopping local services
- Runtime framework for deploying configurations via components
- Centrally-controlled fabric management: configuration server defines the required fabric configuration
- Bare metal installation
- Currently Linux-oriented

Overview of LCFG

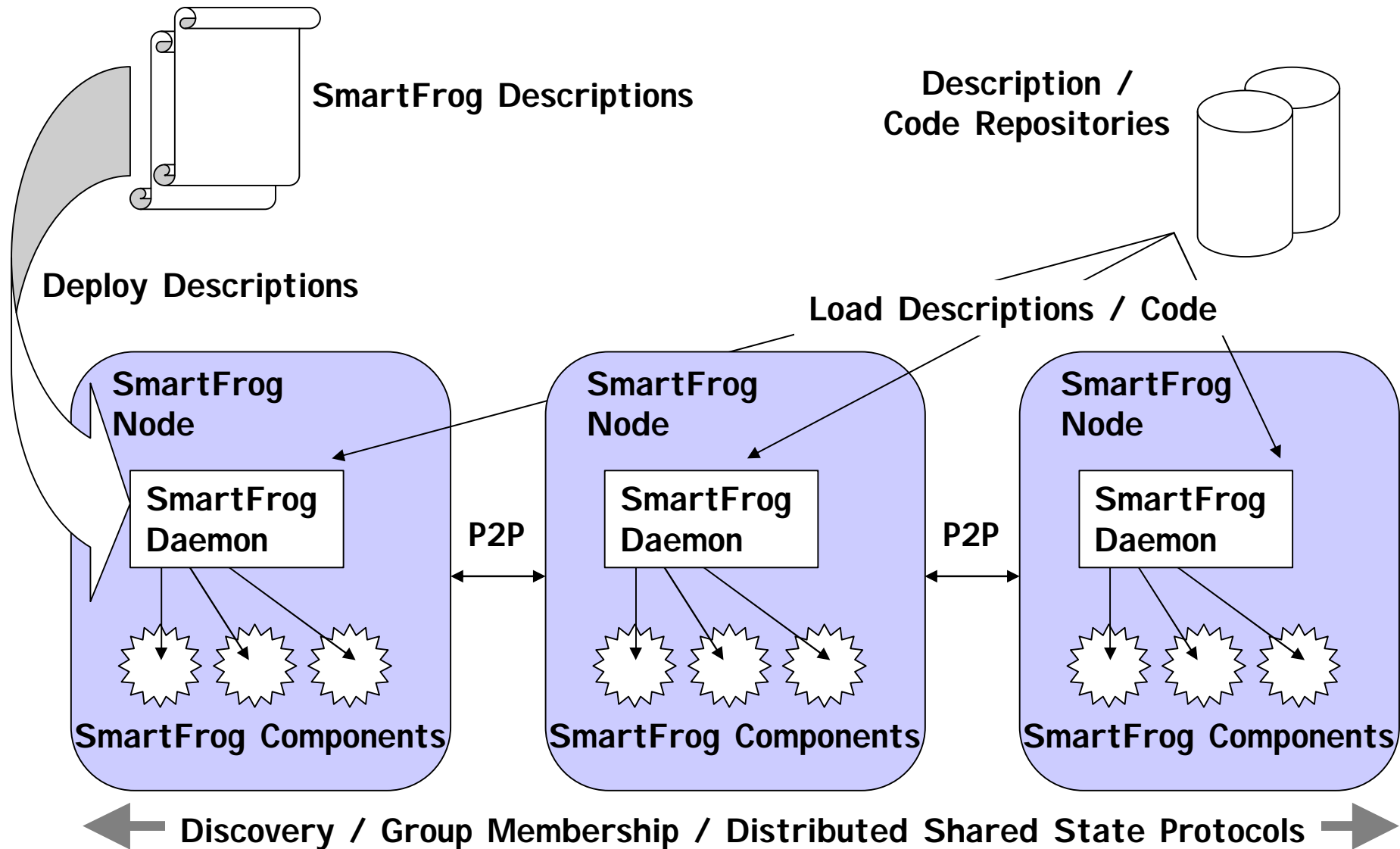


Overview of SmartFrog

SmartFrog: “Smart Framework for Object Groups”

- Describes, deploys and orchestrates distributed, component-based applications
- Language
 - Declarative, prototype-based, attribute description language, supporting templates, composition, late attribute binding, predicates (soon), etc.
- Distributed, runtime deployment infrastructure
 - Secure (certificate-based) deployment of descriptions and code
 - Multiple methods of loading descriptions and code
- Component model and configurator components
 - A defined component lifecycle for each component
 - Extensible set of components developed for grouped lifecycles, workflows, flexible binding mechanisms, etc.
- No central point of control required; peer-to-peer interactions
- Portable across many classes of fabric element (written in Java)

Overview of SmartFrog

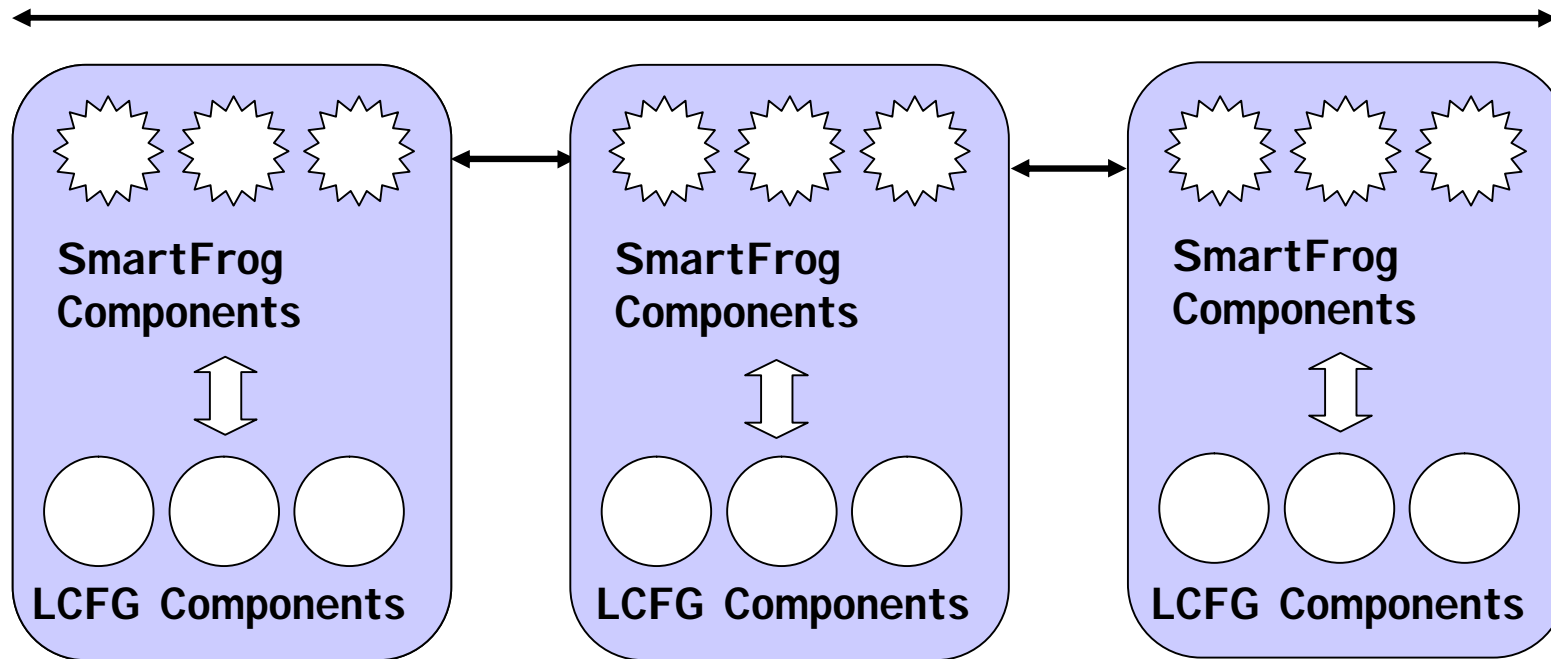


Combining LCFG and SmartFrog

- Division of labour
 - LCFG: per-node configuration, bare-metal upwards
 - SmartFrog: higher-level, distributed, adaptive services
- Integrated configuration infrastructure
 - LCFG configures, starts and manages the SmartFrog daemon on each node
 - SmartFrog controls LCFG components using a generic SmartFrog / LCFG adaptor
- Planned (but not yet done): unified description language (using SmartFrog language v2)
 - Complete fabric description using one, powerful representation
 - “Compilation” results in LCFG node configuration profiles plus deployable SmartFrog descriptions

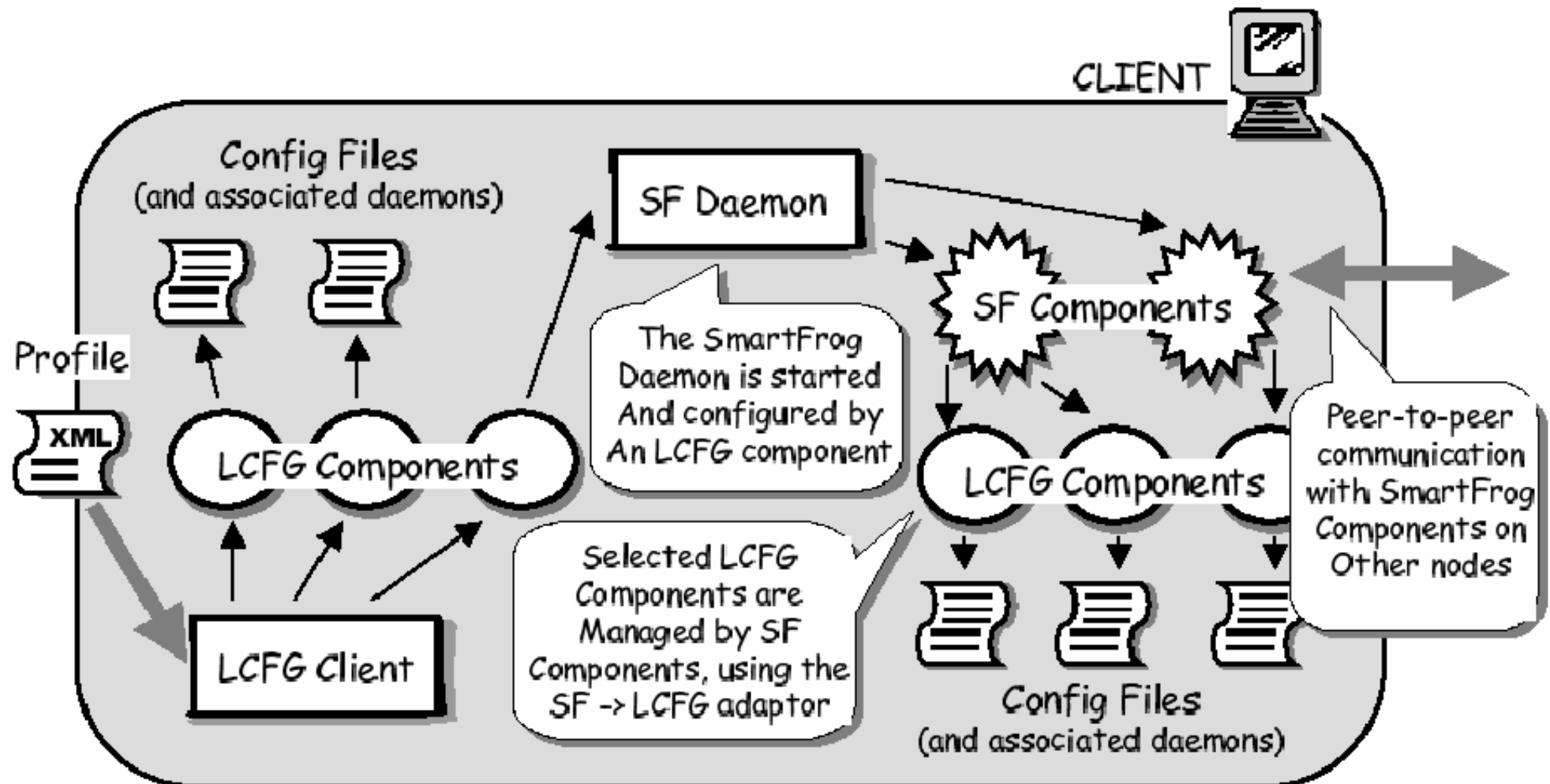
Combining LCFG and SmartFrog

SmartFrog manages distributed, adaptive services across nodes in the fabric



LCFG server configures and manages each node in the fabric, and starts the SmartFrog infrastructure (including initial descriptions)

Combining LCFG and SmartFrog

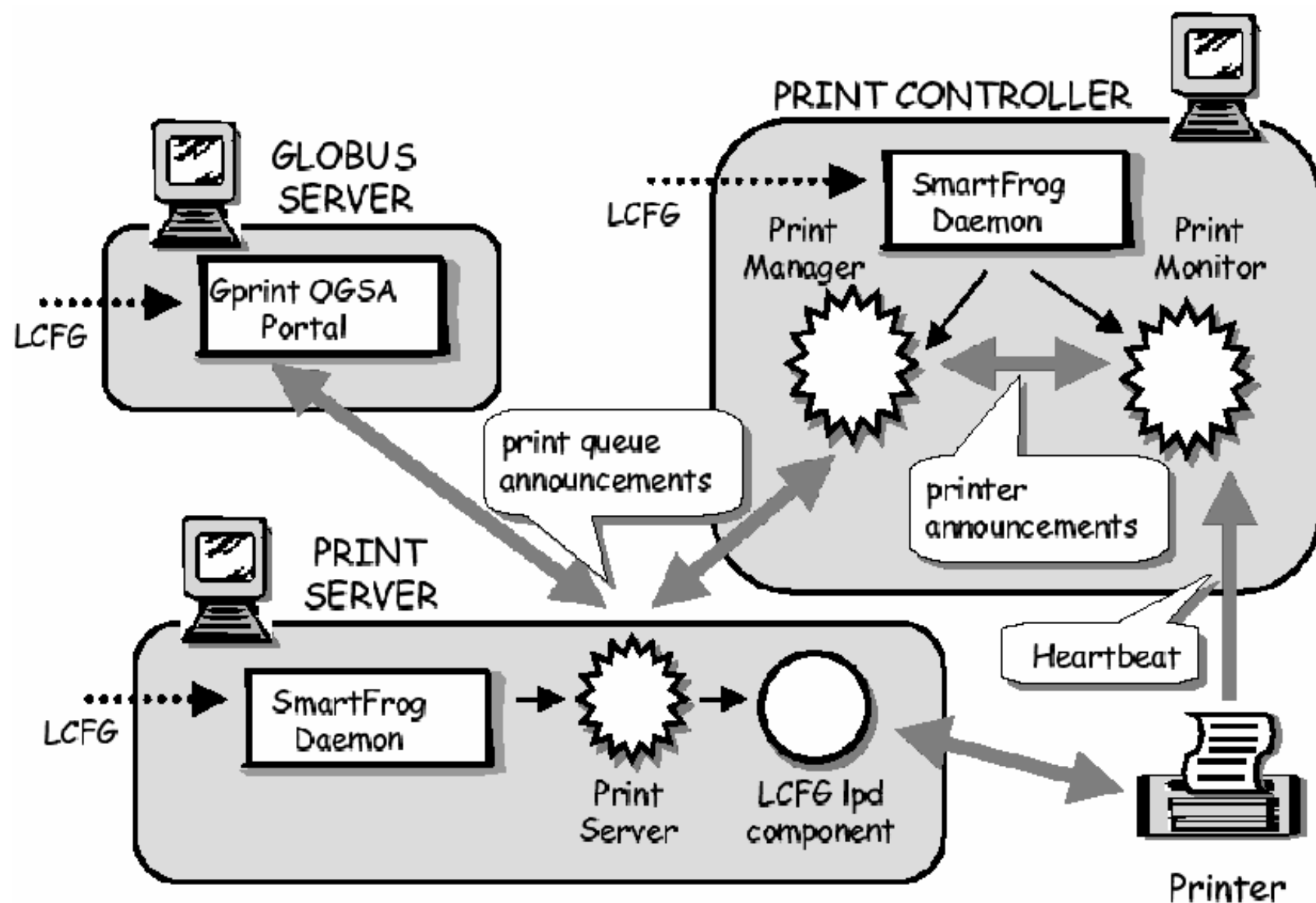


GridWeaver Prototype

- Bare-metal OS installation, configuration and ignition
 - ? Basic fabric management
- Automatic installation and configuration of a Globus GT3 (OGSI) infrastructure
 - ? Grid-enabled fabric
- GPrint: an example adaptive, cross-fabric print service
 - ? Fabric + adaptive service
- Exposure of GPrint as a Grid Service via Globus
 - ? Grid-enabled service



GPrint Service Architecture



Video Sequence



Future Work?

- From a research prototype to a production system
 - LCFG is open source (GPL)
 - SmartFrog release to open source (LGPL) in '03
 - Work needed to create a portable, production-quality system

- Grid Configuration Service Interface
 - A Grid Service interface for fine-grained control over aspects of fabric configuration

- Many other interesting research problems
 - Representing time-based properties in configurations
 - Devolving control of different configuration aspects, securely
 - ...

Project Reports

Technologies for Large-Scale Configuration Management

Paul Anderson - panderson@inf.ed.ac.uk
George Beckett - g.beckett@epic.ed.ac.uk
Kostas Katsimantas - k.katsimantas@epic.ed.ac.uk
Guillaume Mechenieau - guillaume.mechenieau@epic.com
Peter Tobi - peter.tobi@epic.com

Abstract

This report examines the current state-of-the-art in large-scale system configuration. It introduces the scope and principles of the configuration task, and describes a selected range of technologies that are representative of the approaches in use today. It concludes by summarising the dominant approaches to system configuration offered by these technologies.

Acknowledgements

The report is a deliverable of the GridWeaver project, a collaboration between the University of Edinburgh School of Informatics, EPIC Ltd in Bristol, and EPIC. GridWeaver is part of the UK e-Science Grid Programme, and we gratefully acknowledge the programme's assistance with this activity. We also wish to thank the other members of the GridWeaver team – Carolyn Edwards, Patrick Guldback, John Handley – for their contributions to the project as well as Alexander Hill for his valuable input. For administrative purposes, GridWeaver is also known under the project name EP-Flightline.



Revision 1.0 – December 9, 2002

Report 1: “Technologies for Large-Scale Configuration Management”

December 2002

Report 2: “Experiences and Challenges of Large-Scale System Configuration”

March 2003

Experiences and Challenges of Large-Scale System Configuration

Paul Anderson - panderson@inf.ed.ac.uk
George Beckett - g.beckett@epic.ed.ac.uk
Kostas Katsimantas - k.katsimantas@epic.ed.ac.uk
Guillaume Mechenieau - guillaume.mechenieau@epic.com
John Peterson - jpeterson@inf.ed.ac.uk
Peter Tobi - peter.tobi@epic.com

Abstract

This report identifies the large-scale fabric configuration requirements of the emerging Grid world. We present a number of real-world case studies to illustrate how several different organisations currently manage large-scale system fabrics. We then examine some proposed use cases that aim to predict the way in which large-scale system configuration will need to evolve in order to meet the challenges of tomorrow's fabrics. We conclude the report with a summary of key research challenges.



Revision 0.3 (Draft) – March 20, 2003

More Information

○ Contacts

- HP: Peter Toft (peter.toft@hp.com)
- University of Edinburgh: Paul Anderson (dcspaul@inf.ed.ac.uk)

○ Web pages

- GridWeaver: www.gridweaver.org
- SmartFrog: www-uk.hpl.hp.com/smartfrog
www.smartfrog.org
- LCFG: www.lcfg.org

○ Paper in LISA '03: “SmartFrog meets LCFG: Autonomous Reconfiguration with Central Policy Control”

○ Further project reports available in August '03 (covering modelling and language, integration architecture, prototype design)