

SmartFrog meets LCFG Autonomous Reconfiguration with Central Policy

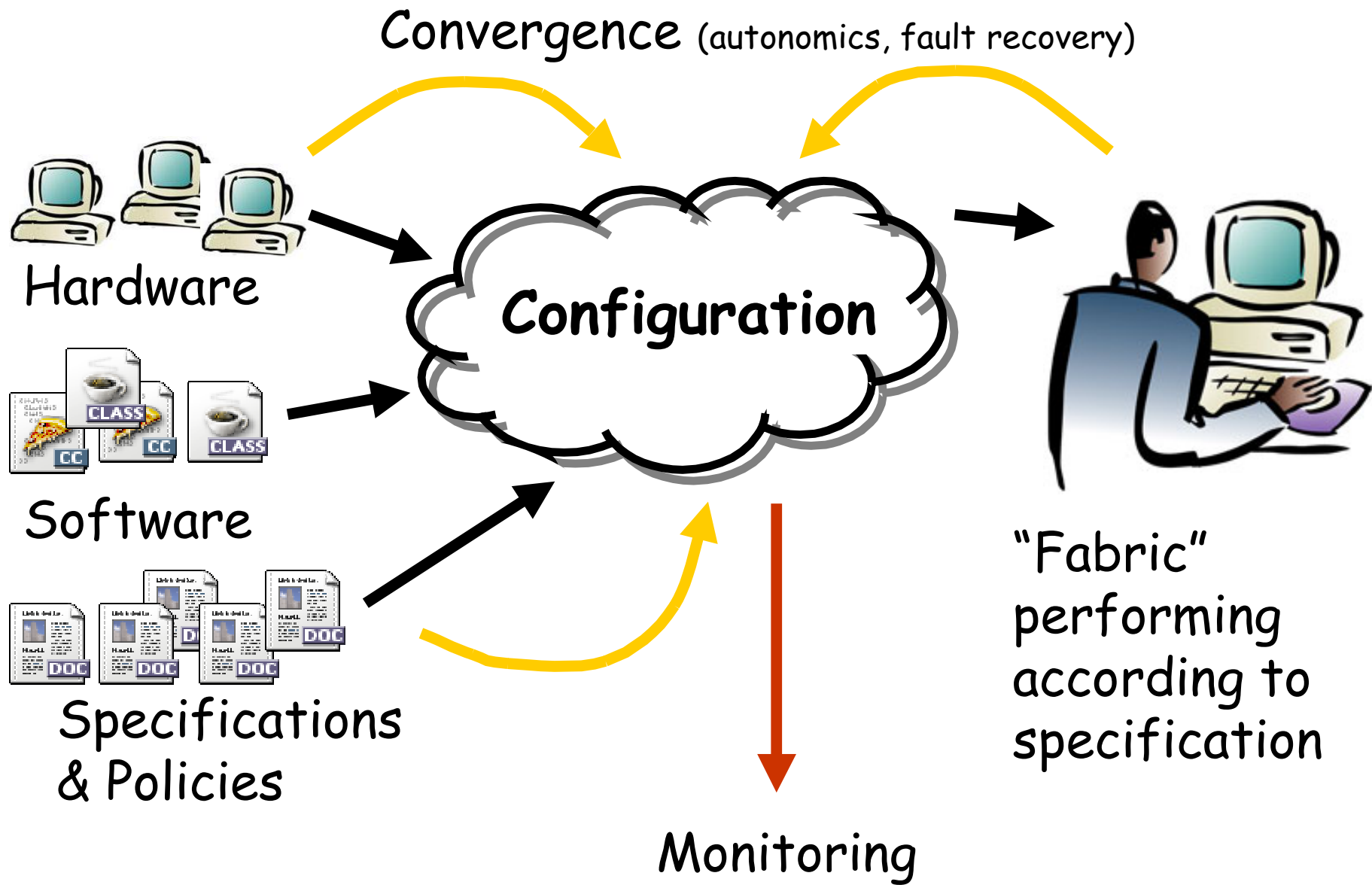
Paul Anderson
<dcspaul@inf.ed.ac.uk>

University of Edinburgh

SmartFrog meets LCFG



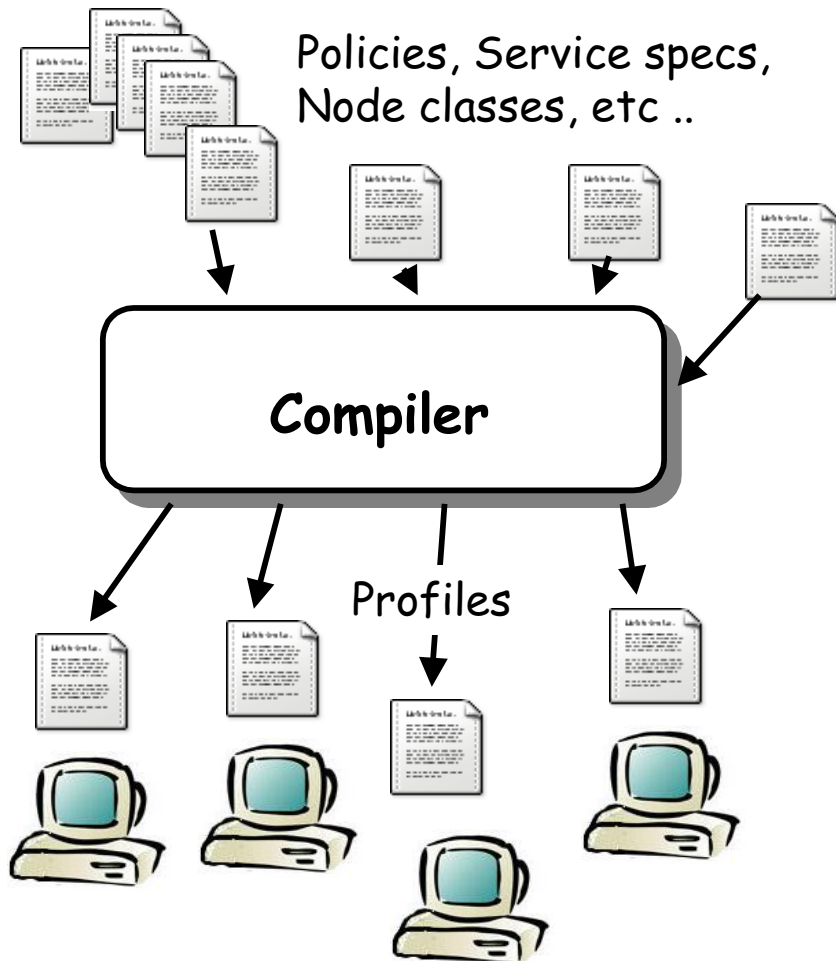
- What is this thing called "system configuration" ?
- What is LCFG?
- What is Autonomous reconfiguration, and why do I want it?
- What is SmartFrog, and how does it help?
- What can we do with it?



The configuration problem

- Starting with:
 - Several hundred new PCs with empty disks
 - A Repository of all the necessary software packages
 - A specification of the required service
- Load and configure the software
 - Including relationships between individual nodes
- Reconfigure the machines when the required service specification changes
- Reconfigure the machines to keep satisfying the specification when the fabric changes
 - Hardware fails, or is replaced with different hardware
- Inform the manager when the specification cannot be satisfied (monitoring)

LCFG is

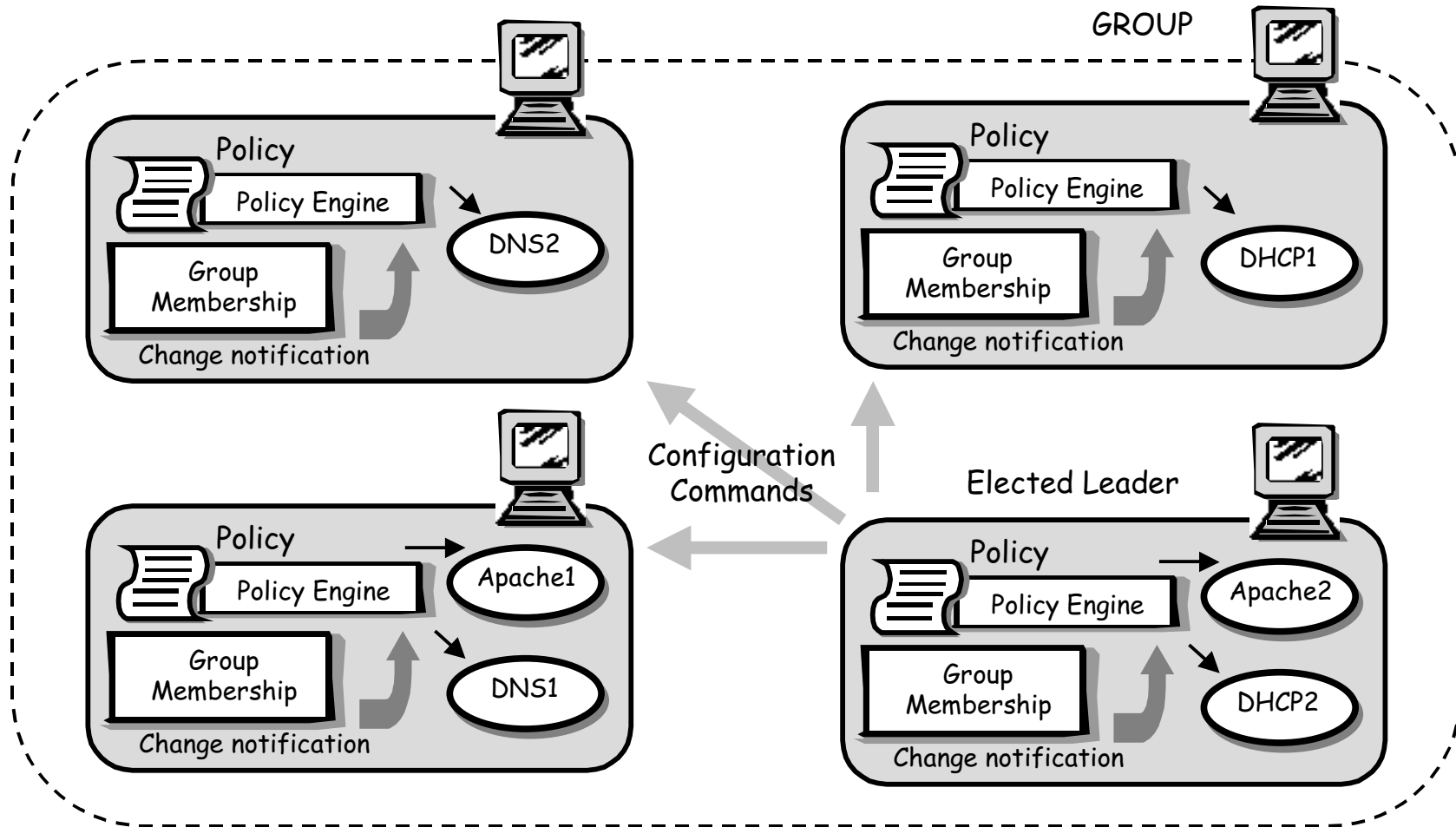


- A "compiler" that takes a relatively high-level specification of the configuration of a complete site and generates one "profile" for each machine
- A client that runs on each node to fetch profiles from the server
- A set of "components" that generate configuration files from the profiles
- LCFG can rebuild a complete site from scratch given the source specification and a repository of software packages

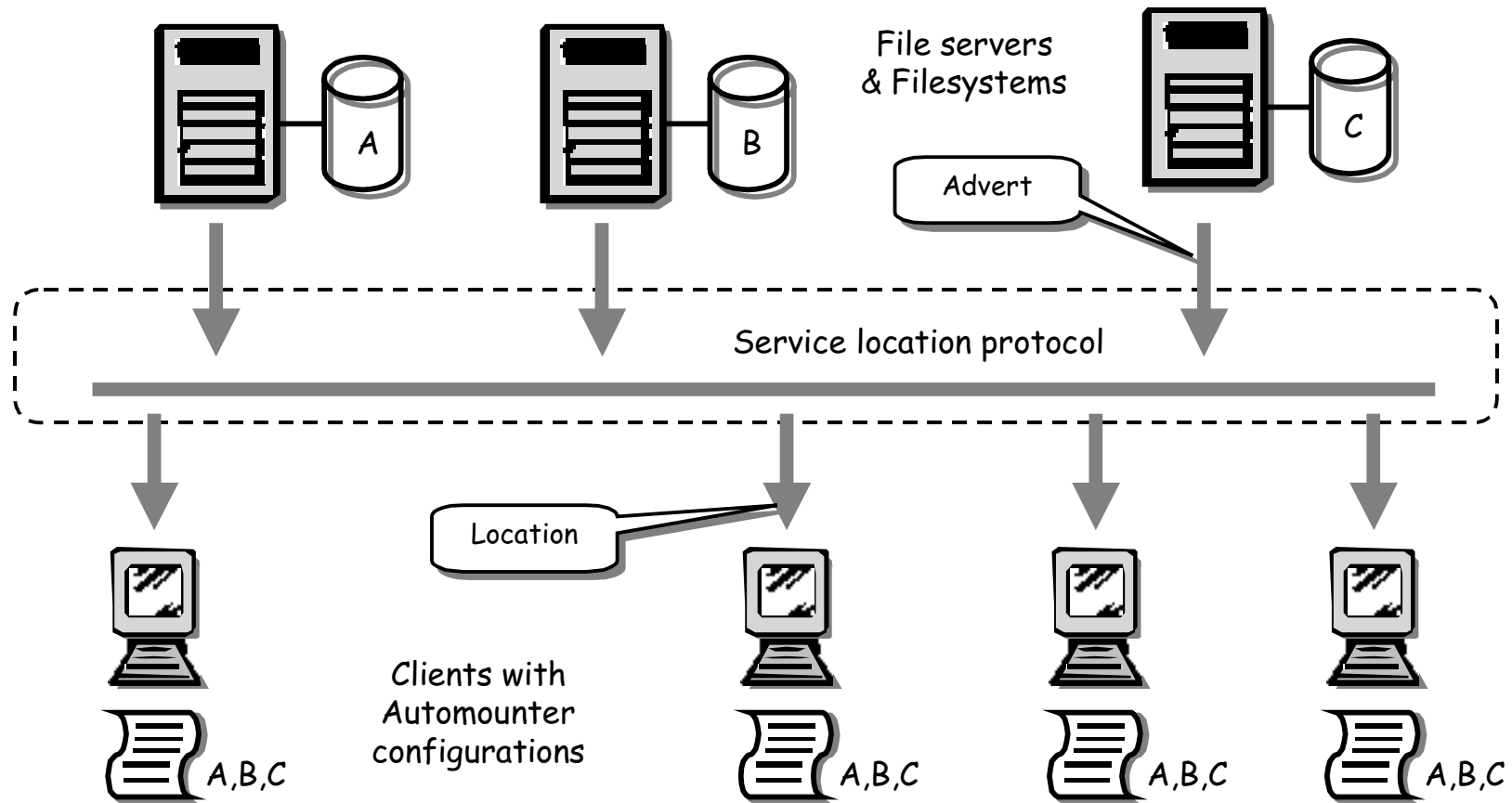
Autonomous reconfiguration

- All configuration changes in LCFG (and similar systems) involve a change to the specification on the central server, and a recompilation of the configuration
- This is not suitable for transient configuration changes, and is ultimately not scalable
- In theory, some aspects of the configuration can be delegated to small groups of machines
 - A group of machines could be allowed to decide amongst themselves which one should provide DHCP service for the group
 - If the DHCP server fails, the group could reconfigure automatically to use a different server without central reconfiguration
- Standard P2P protocols are available which can be used to implement this type of functionality ...

Service reliability



Service location



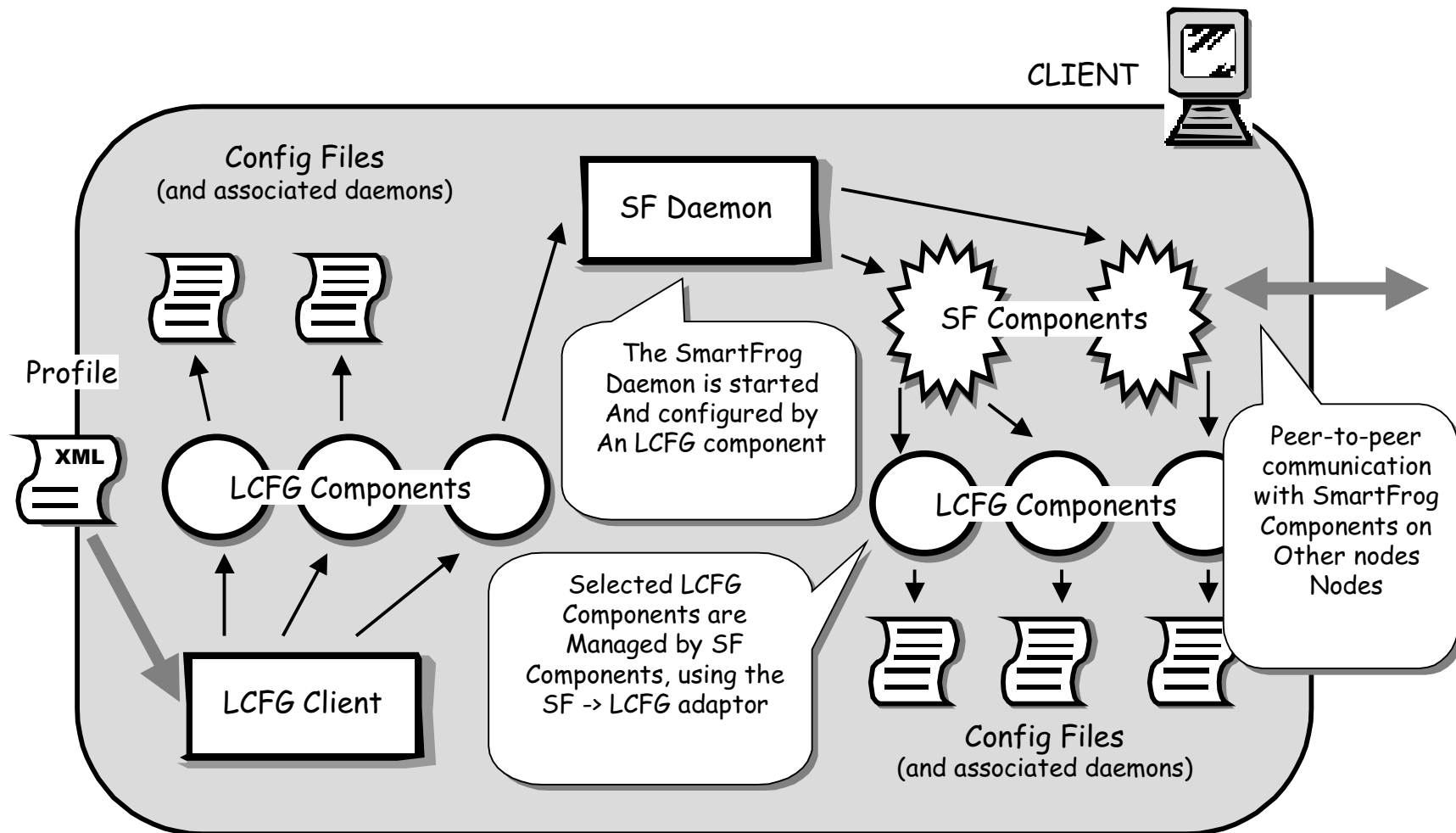
SmartFrog is ...

- A component framework for configuring and managing distributed applications
- It provides a declarative configuration language, and ..
- A runtime environment which “orchestrates” the workflow of the components according to the specified configuration
- It has generic components for implementing useful standard protocols, such as service location and reliability

Autonomous Reconfiguration with Central Policy Control

- By combining LCFG with SmartFrog, we can ...
- Use existing LCFG components to deploy the configuration, but configure them either from the central specification, or dynamically via SmartFrog
- The central configuration specification controls the SmartFrog system itself and defines how each component is configured
- This allows us implement, and experiment with, autonomous reconfiguration of various subsystems with very little additional code and effort

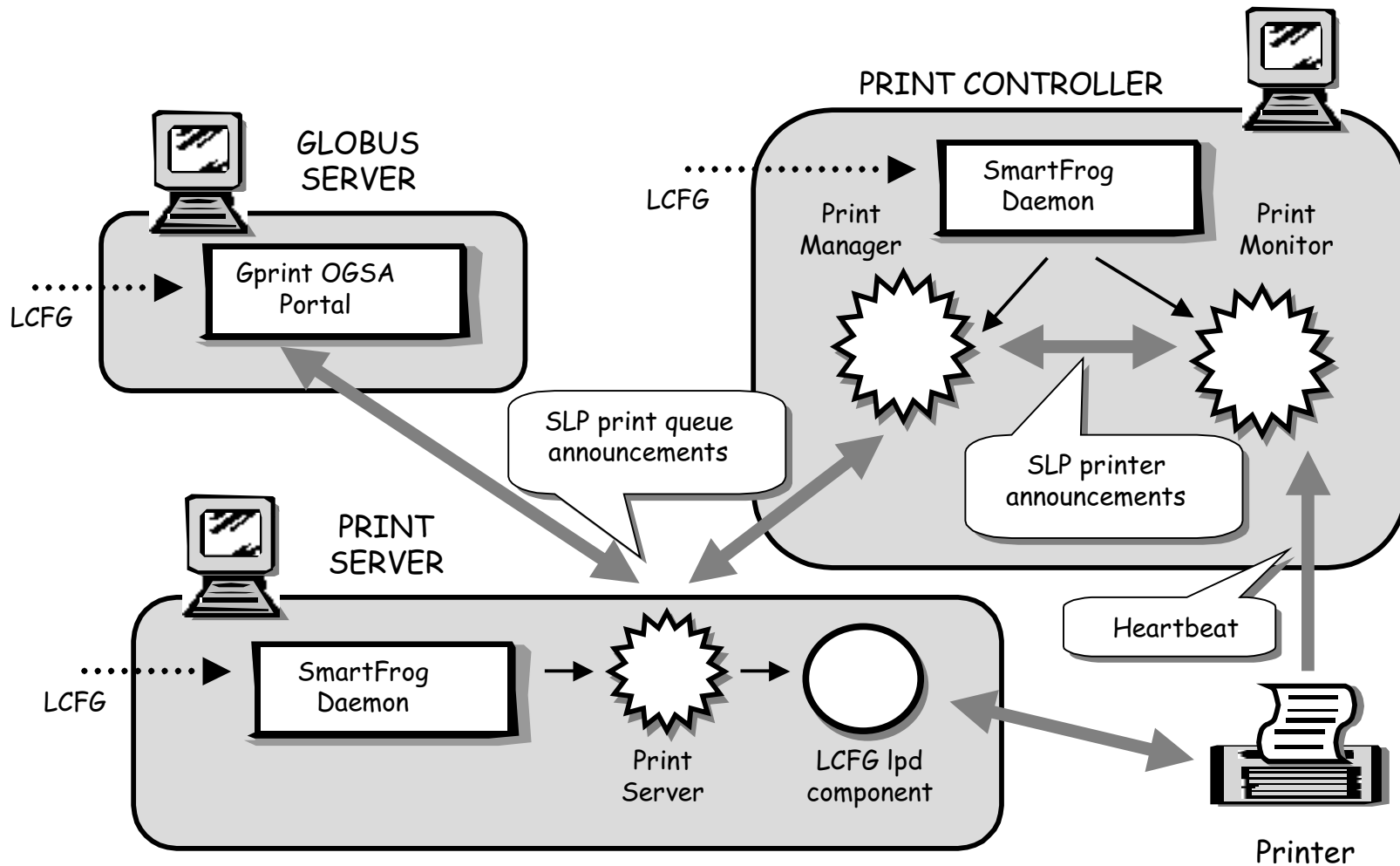
A combined framework



GPrint

- As part of the *GridWeaver* project, we built a demonstration printing service using this framework ...
- This is based on our production LCFG/LPRng technology
- The central configuration specifies the machines which are eligible to be print servers
- Printers are detected automatically and allocated to print servers, balancing the load
- If a new print server appears, queues are reallocated
- If a print server dies, the queues for its printers are transferred to another server
- If a printer dies, its jobs are transferred to another queue

The GPrint architecture



Conclusions

■ GPrint only a demonstrator!

- Is is built by putting together two separate technologies with duct tape
 - The technologies currently use separate configuration languages
- The point is
 - It is possible to build a framework where the same components can move between dynamic and static configuration
 - A few simple, generic P2P protocols can be used to build robust self-healing systems, with very little additional code
 - We would like to build a system which supports this in a fully unified way

References



- **LCFG**
<http://www.lcfg.org>
- **SmartFrog**
<http://www-uk.hpl.hp.com/smartfrog>
- **GridWeaver**
<http://www.gridweaver.org>
- **The Movie**
<http://boombox.ucsf.edu/ramgen/informatics/gridweaver.rm>
- .. Or ..
<http://boombox.ucsf.edu/ramgen/informatics/gridweaver-v8.rm>