

Distributing System Configuration

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

<http://homepages.inf.ed.ac.uk/dcspaul>

Overview

- Background
 - System configuration
 - Current solutions
 - Problem areas
- Distributed approaches
 - Ad-hoc distributed techniques
 - A more general approach
 - The distributed composition problem

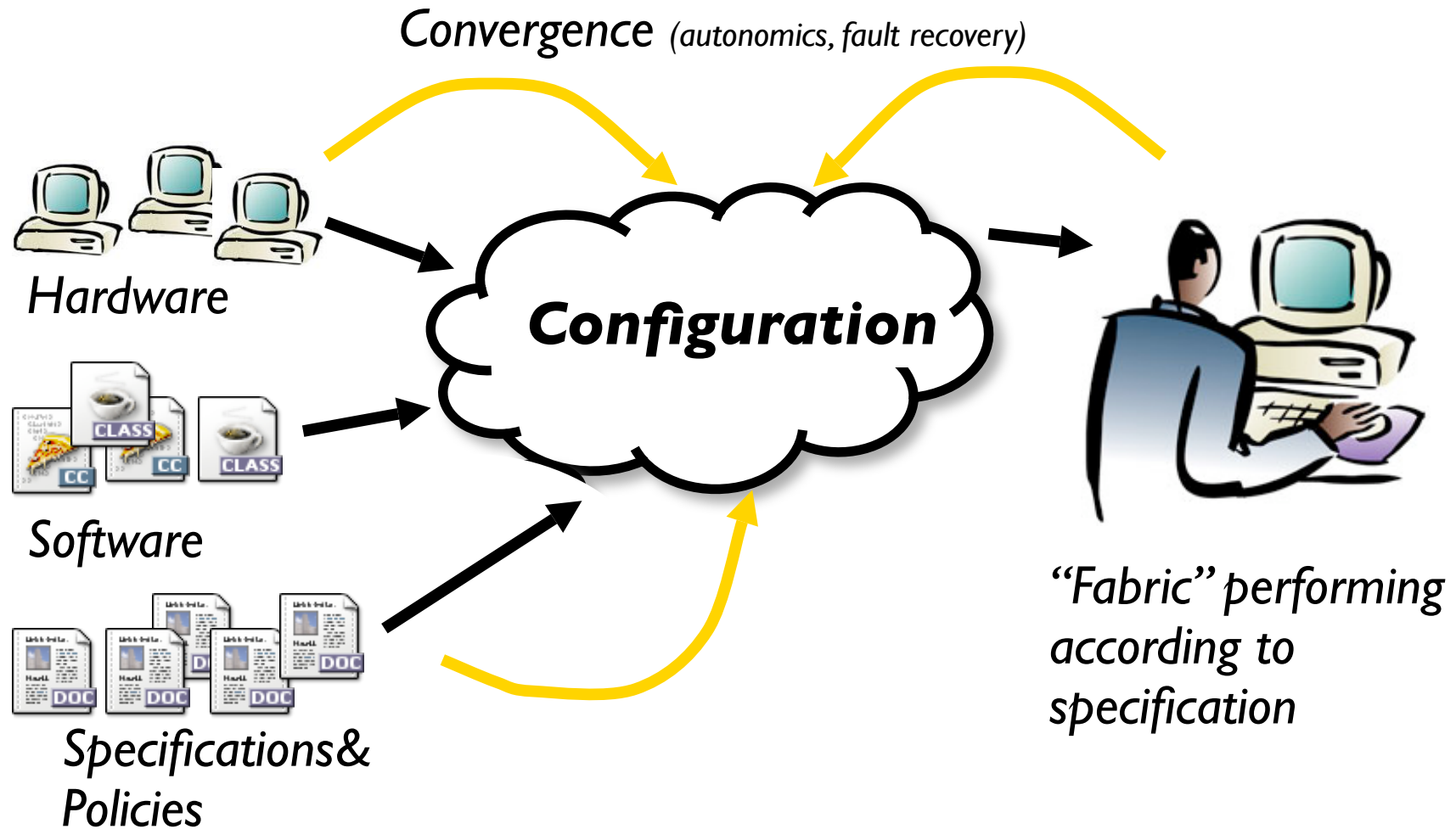
System Configuration

- Starting with:
 - Several hundred new machines with empty disks
 - A Repository of all the necessary software packages
 - A specification of the required service
- Load the software and configure the machines to provide the service
- This involves many internal services:
 - DNS, LDAP, DHCP, NFS, NIS, SMTP, Web ...
 - The relationships are most important

Change and Diversity

- Modern installations change rapidly:
 - Reconfigure the machines when the required service specification changes
 - Reconfigure the machines to keep providing the specified service when the fabric changes
- Modern installations are very diverse:
 - At least the “servers” are diverse

System Configuration



Automation - why?

- Efficiency - obviously
- Manage the complexity
- Have confidence in the correctness
 - What happens if one machine in 1000 node cluster has the wrong version of a maths library?
- Security
 - Can we be confident there are no insecure dependencies?
- Reliability (autonomic reconfiguration)

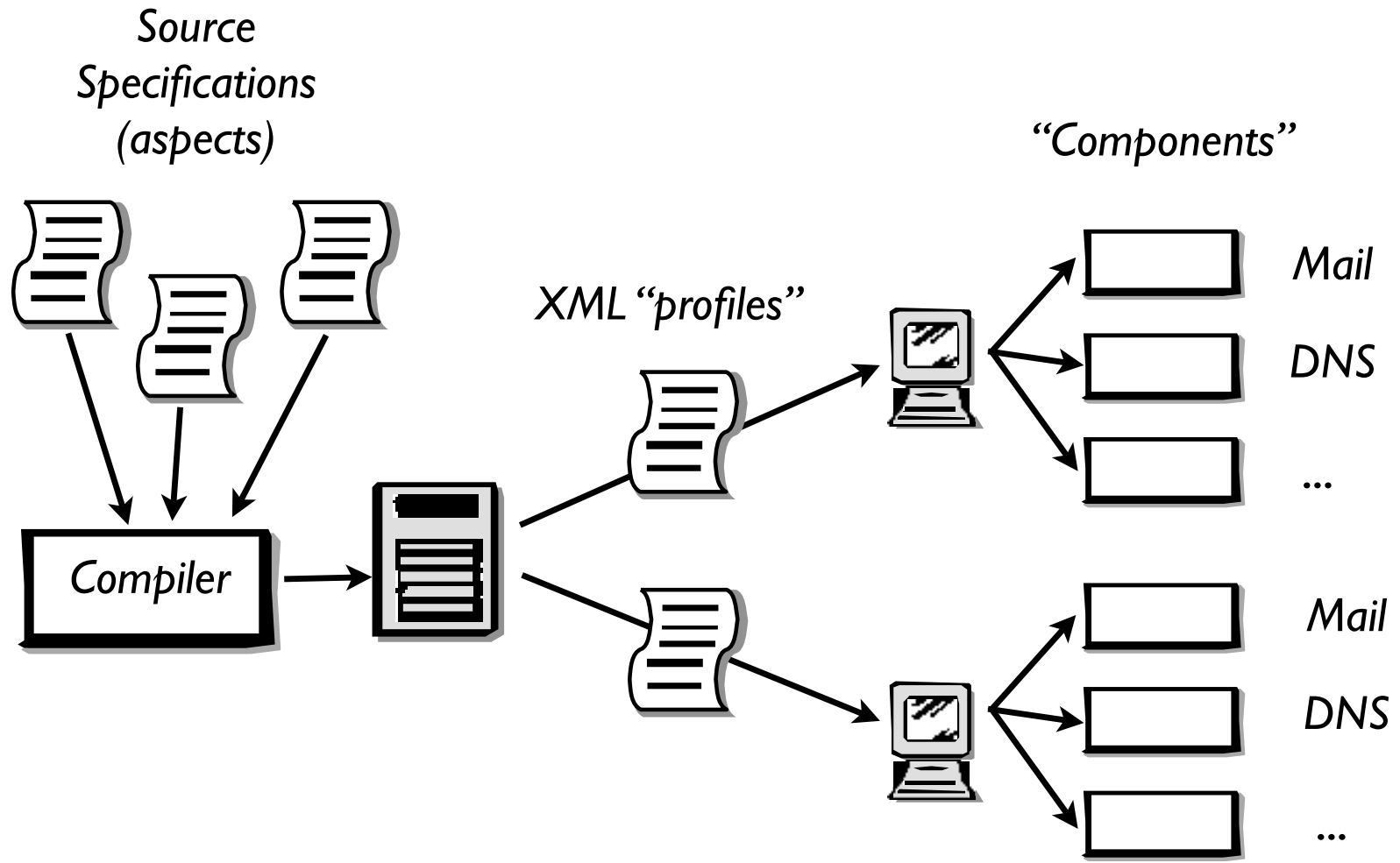
Some considerations

- The behaviour of the entire fabric is the important issue (not individual nodes)
- Many different people are involved in constructing different “aspects” of the specification
- In reality, configuration is “asymptotic”
- Unpredictable and unreliable behaviour is normal
- Awkward, legacy formats and interfaces are normal
- Most non-trivial configuration tasks are currently highly skilled manual processes

The state of the art

- Current practical systems are centralised:
 - Specifications are assembled on a configuration server which controls the configuration of all the other machines.
- “Higher-level” configuration decisions are made by humans
 - Eg, the physical location of services
- The specifications are “compiled” (composed) into lower-level parameters (resources) which are used to configure the individual machines

LCFG



Some current issues

- Automatic fault recovery (autonomics) requires “higher levels” of automation:
 - Inter-machine autonomics involves the reconfiguration of relationships
- More complex relationships between systems and services requires support for highly devolved configuration:
 - Informatics may devolve the mail setup of machines to EUCS
 - I may devolve the AFS setup of my laptop to Informatics

Some motivation for a more autonomous approach

- Reconfiguration is often necessary to recover from failures, and needs to operate in an uncertain environment
- Reconfiguration needs to be fast, especially when recovering from failures
- The source of the configuration information itself is distributed
 - A system manager
 - Some other human (eg. the user?)
 - Automatically generated information (eg. the state of some other machine)

Dynamic protocols

- There is a wide range of dynamic protocols which can be used to configure different aspects of the system automatically. For example:
 - DHCP allocates IP addresses
 - Zeroconf locates and configures access to print servers (or remote iTunes, or chat, or)
 - uPNP configures firewalls according to service requirements
 - Election protocols ensure a single active server

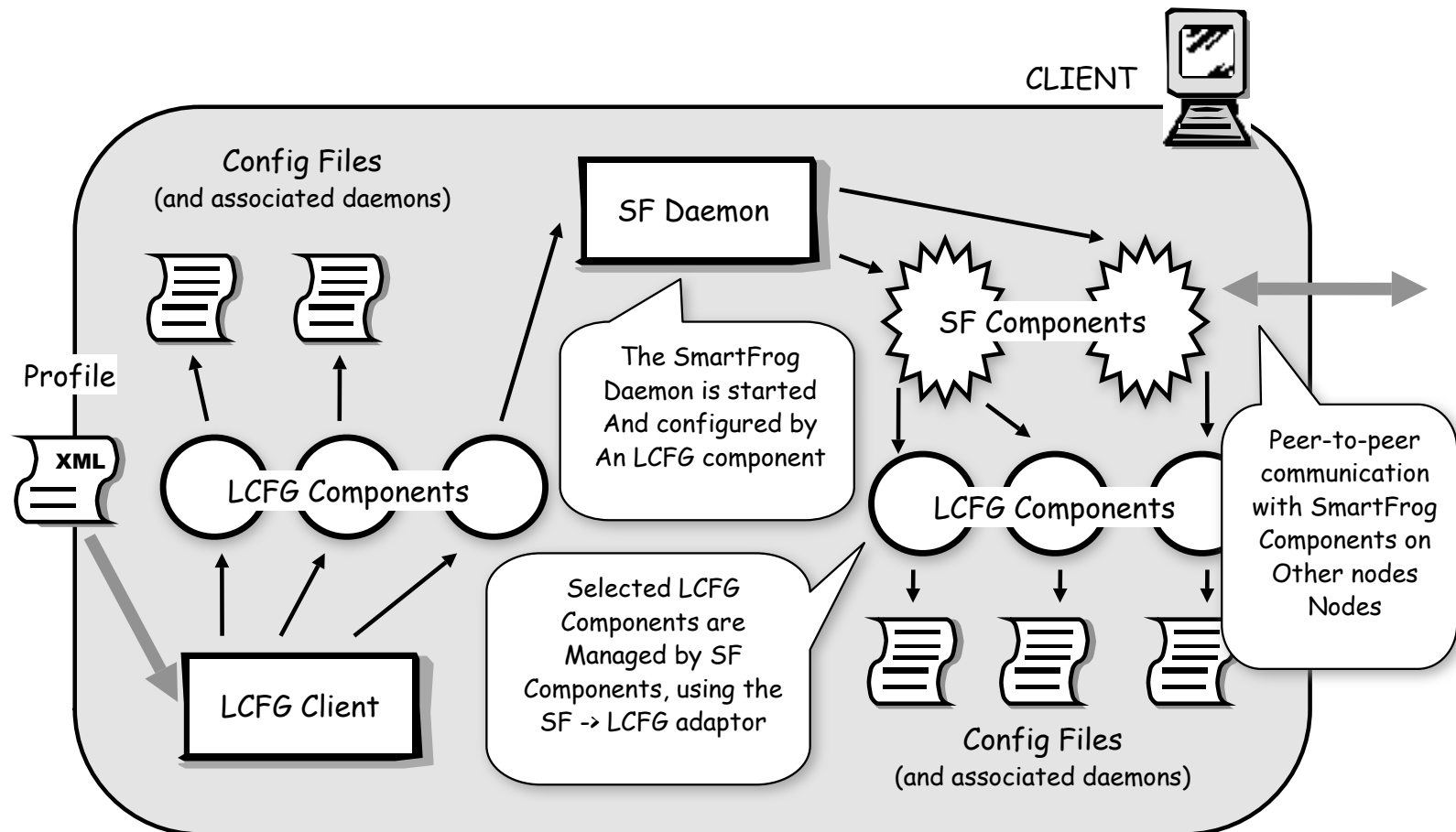
Problems with dynamic configuration

- They use completely different technologies!
- There is no mechanism for configuring the policy
 - Which machines are allowed to request which holes in the firewall?
 - Which machines are candidates for providing a redundant service?
- These protocols operate at different “levels”
- They don't address the distributed composition of requirements

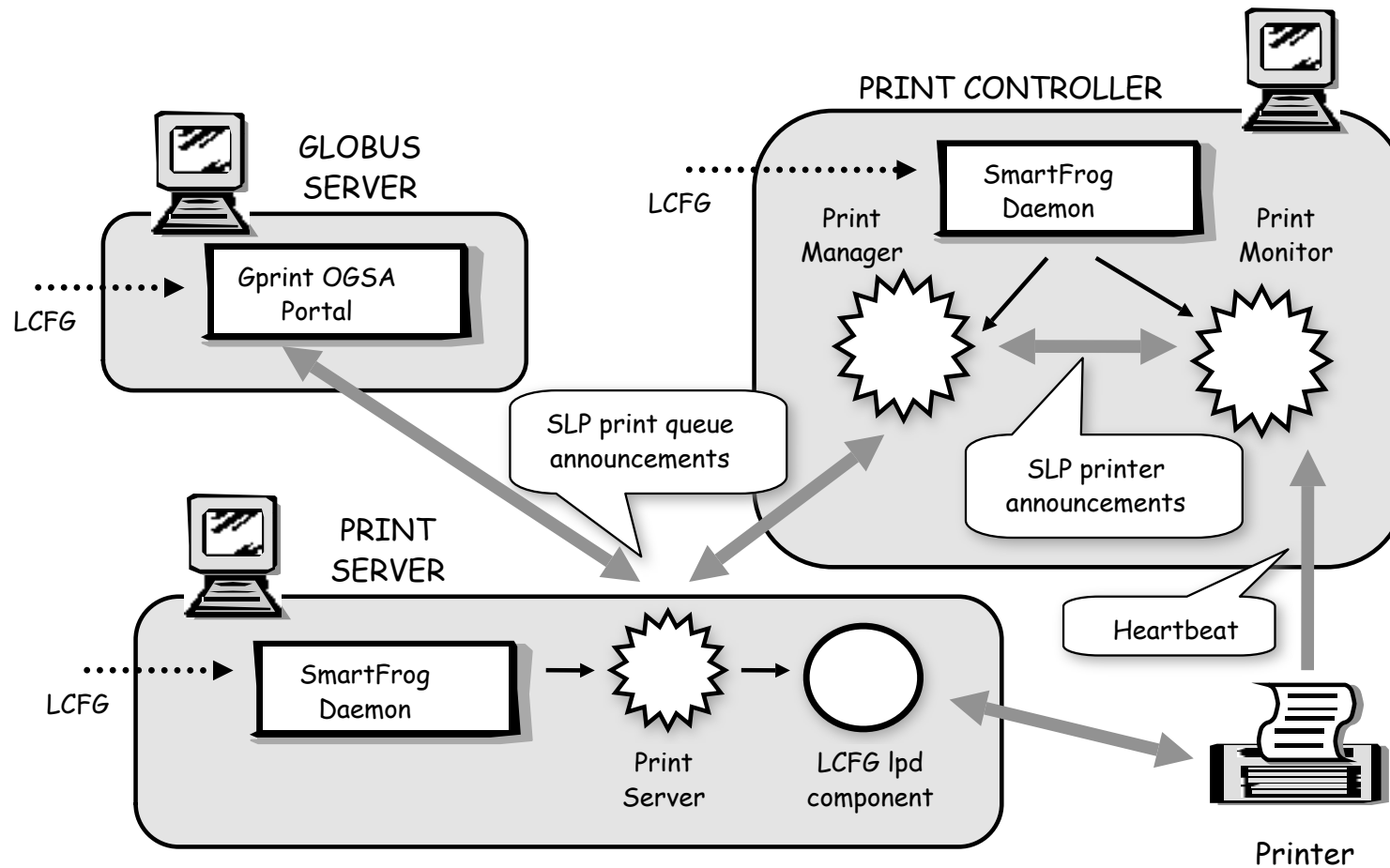
Autonomous reconfiguration with central policy

- The GPrint demonstrator combines:
 - P2P election protocols in the SmartFrog framework
 - Central policy provided from the LCFG configuration system
- This allows print servers to reconfigure autonomously in the event of failure
 - The policy is dictated by the centralised LCFG configuration
 - The reconfiguration will continue to operate if the configuration system is down

The GPrint framework



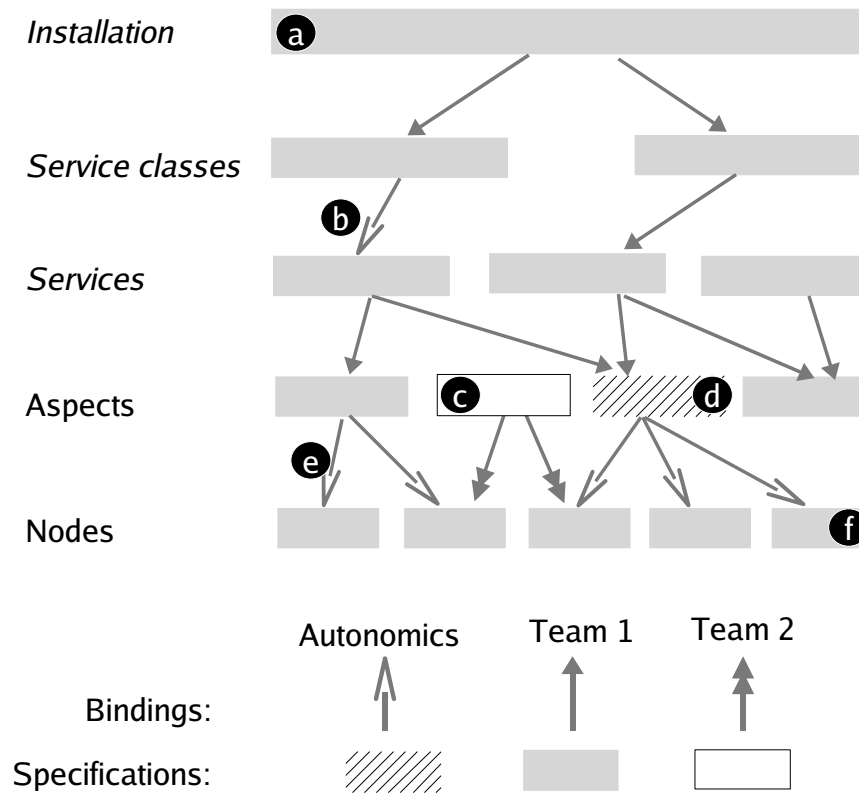
The GPrint architecture



A more general framework

- GPrint is a very ad-hoc implementation where the dynamic component is hand-crafted at a specific “level”
- We can imagine generalising this to a multi-level framework where resources could be injected and transformed at any level, by humans, or autonomic components ...
 - A high-level autonomic component could move services around
 - A low-level autonomic component may reconfigure a failed service on one machine

A “multi-resolution” framework



Generic operations

- Aggregation
- Aspect composition (classing)
- Delegation and authorisation
- Sequencing and planning

Composition

- “Composition” is the way in which resource “outputs” from a level are derived from the set of input resources
- This currently includes simple conflict resolution between values, and set operations
- For more complex autonomies, it needs to support more general constraints
- The format of these is not yet clear:
 - Very general constraints?
 - A very restricted set of operations?

Distributed composition?

- Distributing the multi-resolution model involves distributing the configuration “composition”
- In the most general case, this involves distributed constraint resolution which may be impractical?
- The other extreme is a very restricted set of operations which is difficult to define at this stage
- A flexible framework to experiment with this would be useful
- Perhaps agent technology has a role?

References

- *System Configuration, Paul Anderson*
Short Topics in System Administration, ISBN 1-931971-42-0
- *SmartFrog meets LCFG - Autonomous Reconfiguration with Central Policy Control*
Paul Anderson, Patrick Goldsack and Jim Paterson
Proceedings of the 2003 Large Installations Systems Administration (LISA) Conference
- *Configuration Tools: Working Together*
Paul Anderson and Edmund Smith
Proceedings of the 2005 Large Installations Systems Administration (LISA) Conference
- *Toward Broad-Spectrum Autonomic Management*
Edmund Smith and Paul Anderson
ICONS 2007, to be published
- <http://homepages.inf.ed.ac.uk/dcspaul/publications/cisa2007.pdf>