Configuring & managing the Informatics computing systems

Innovative Learning Week
Wed 17th February 2016

Paul Anderson
Alastair Scobie
Stephen Quinney
Kenny MacDonald

LCFG

Large Scale UNIX Configuration System

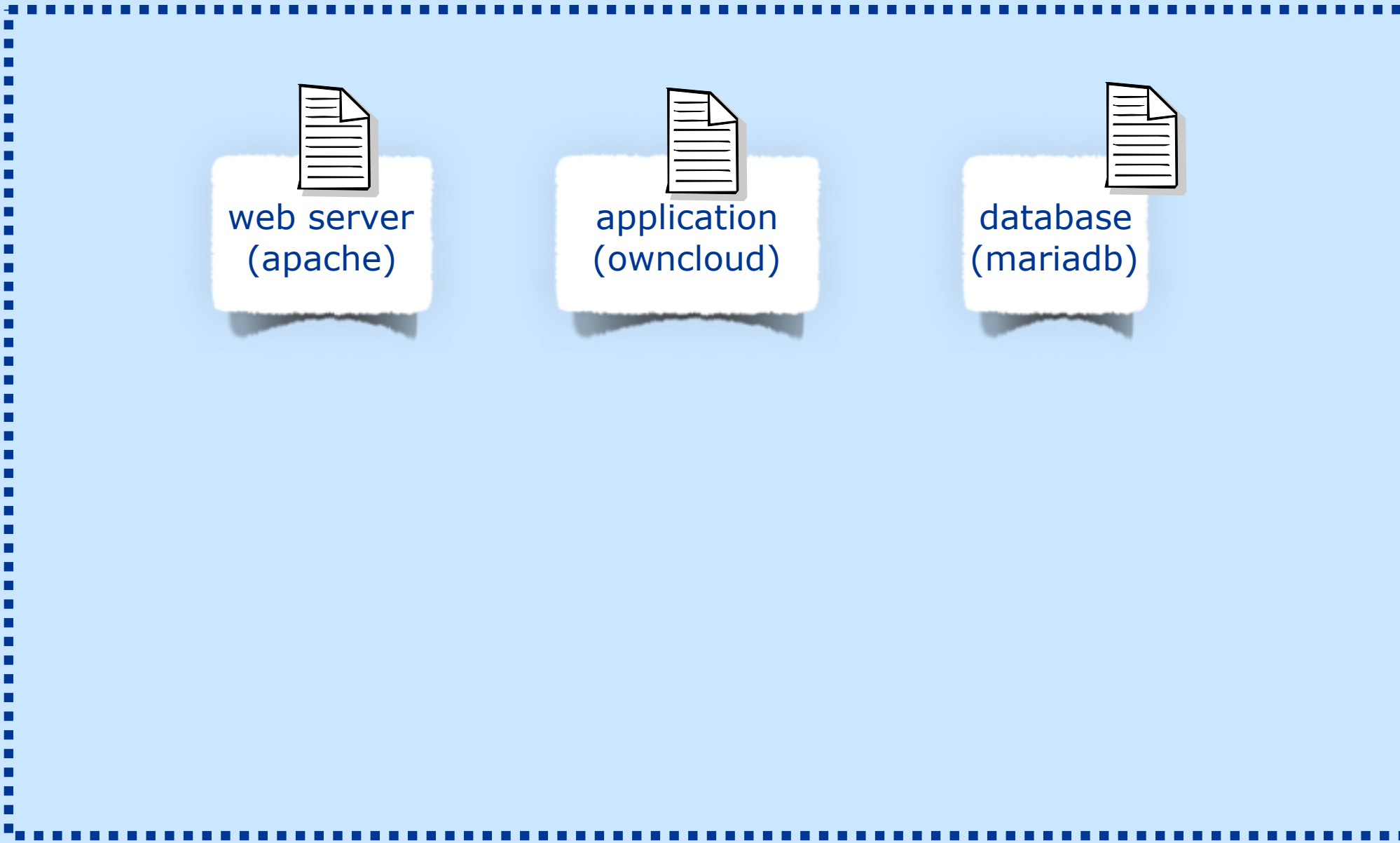
Programme

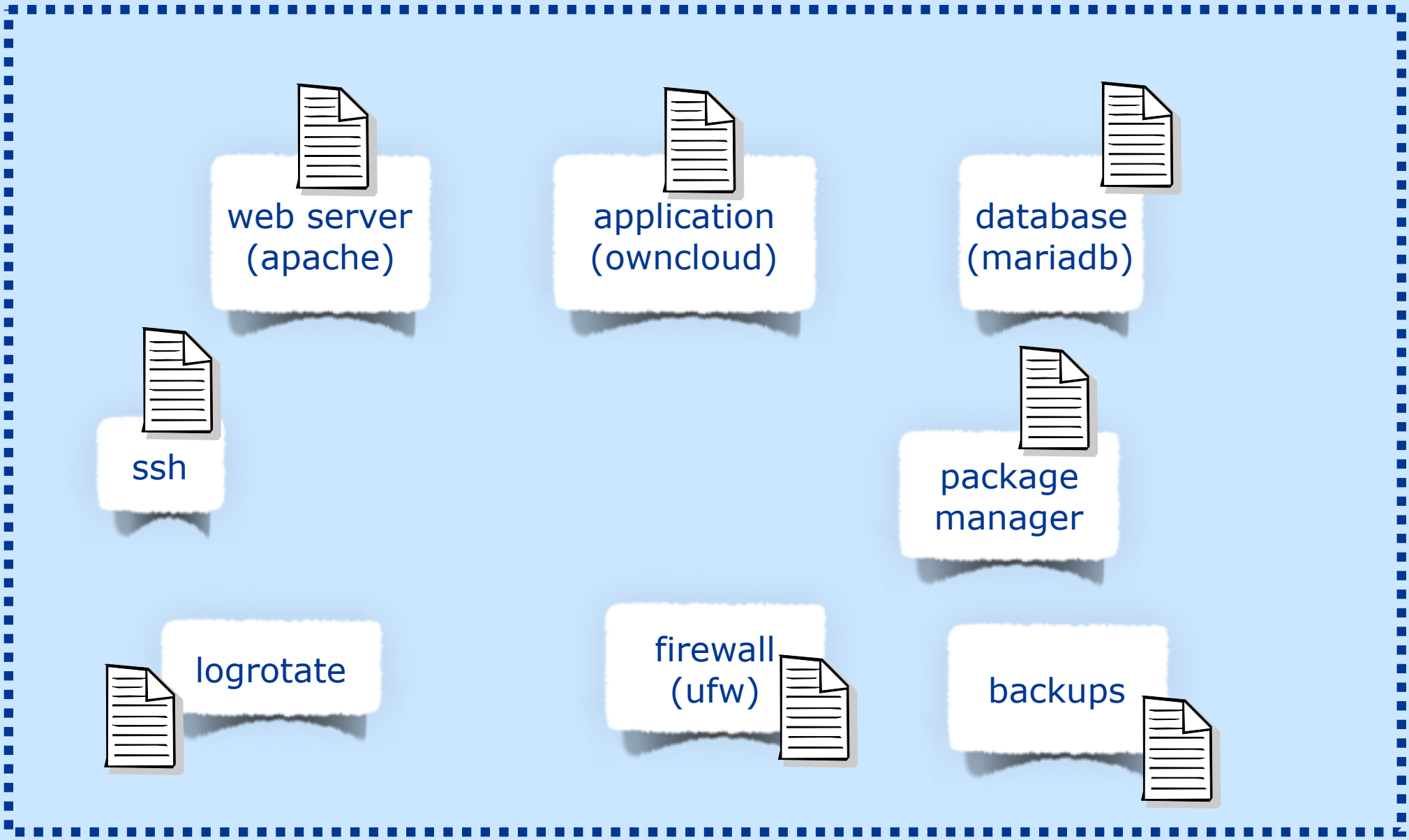
10.00-10.15 - Intro (Paul)

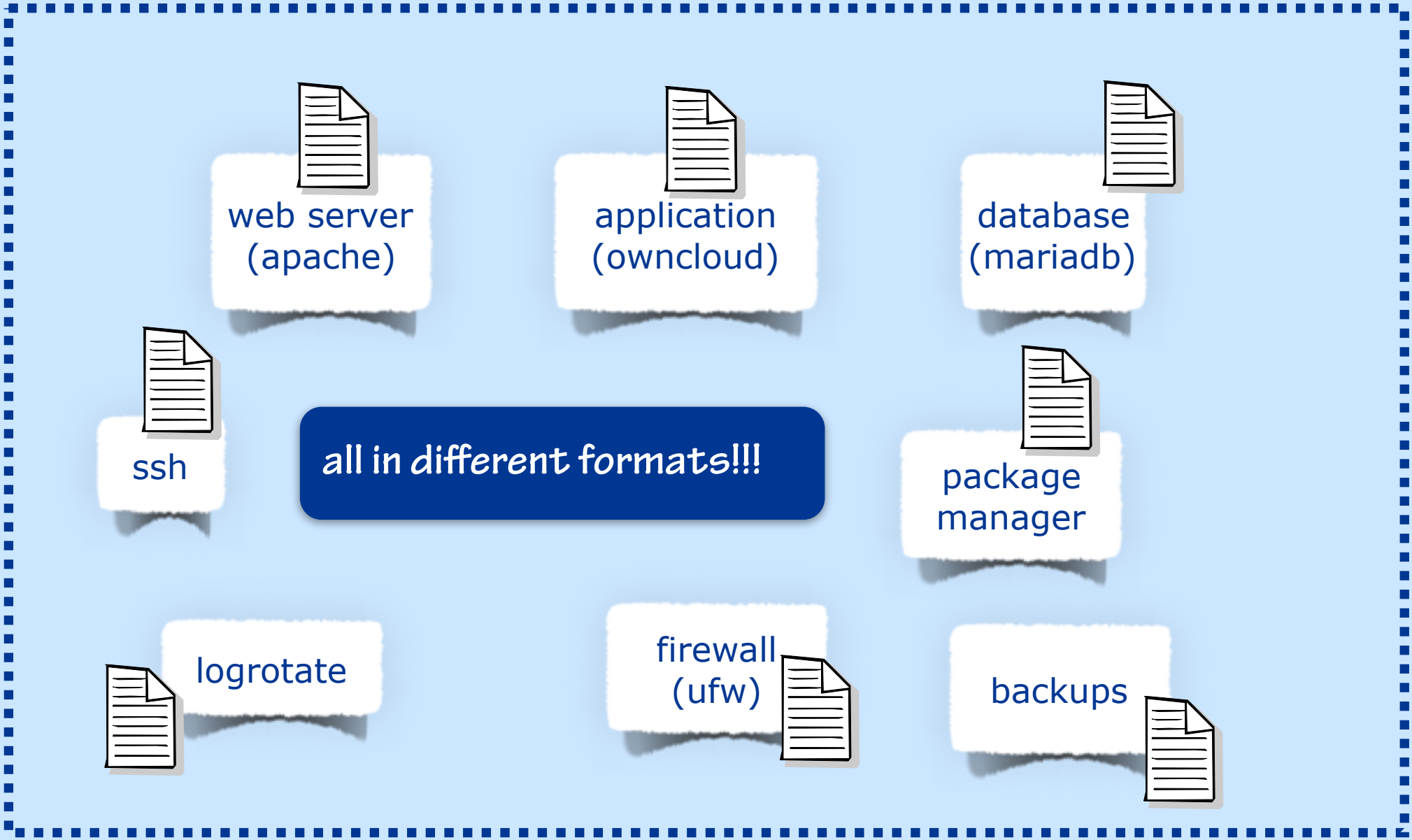
10.15-11.45 - Configuring systems with LCFG (Alastair & Stephen)

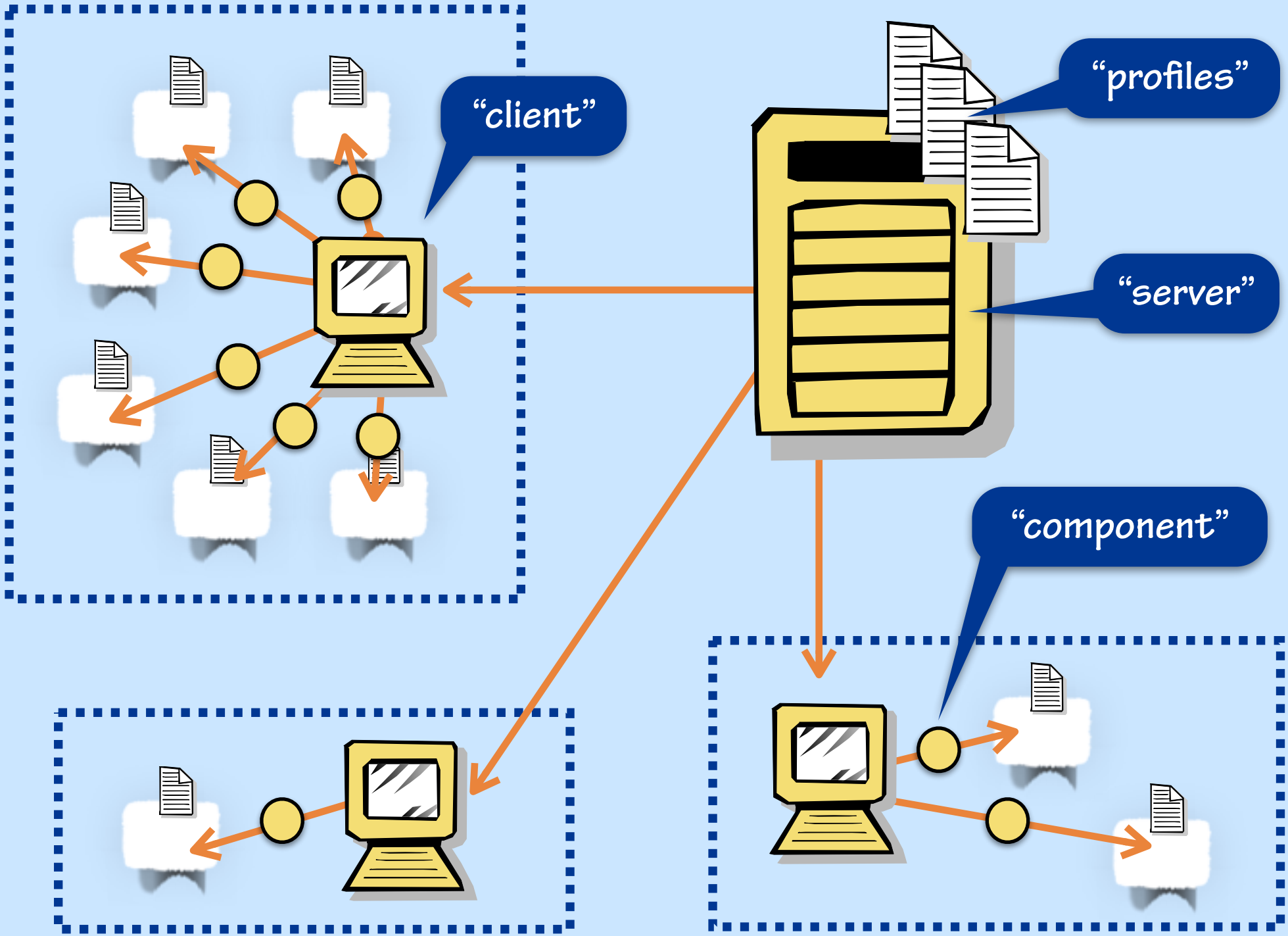
11.45-12.00 - Break

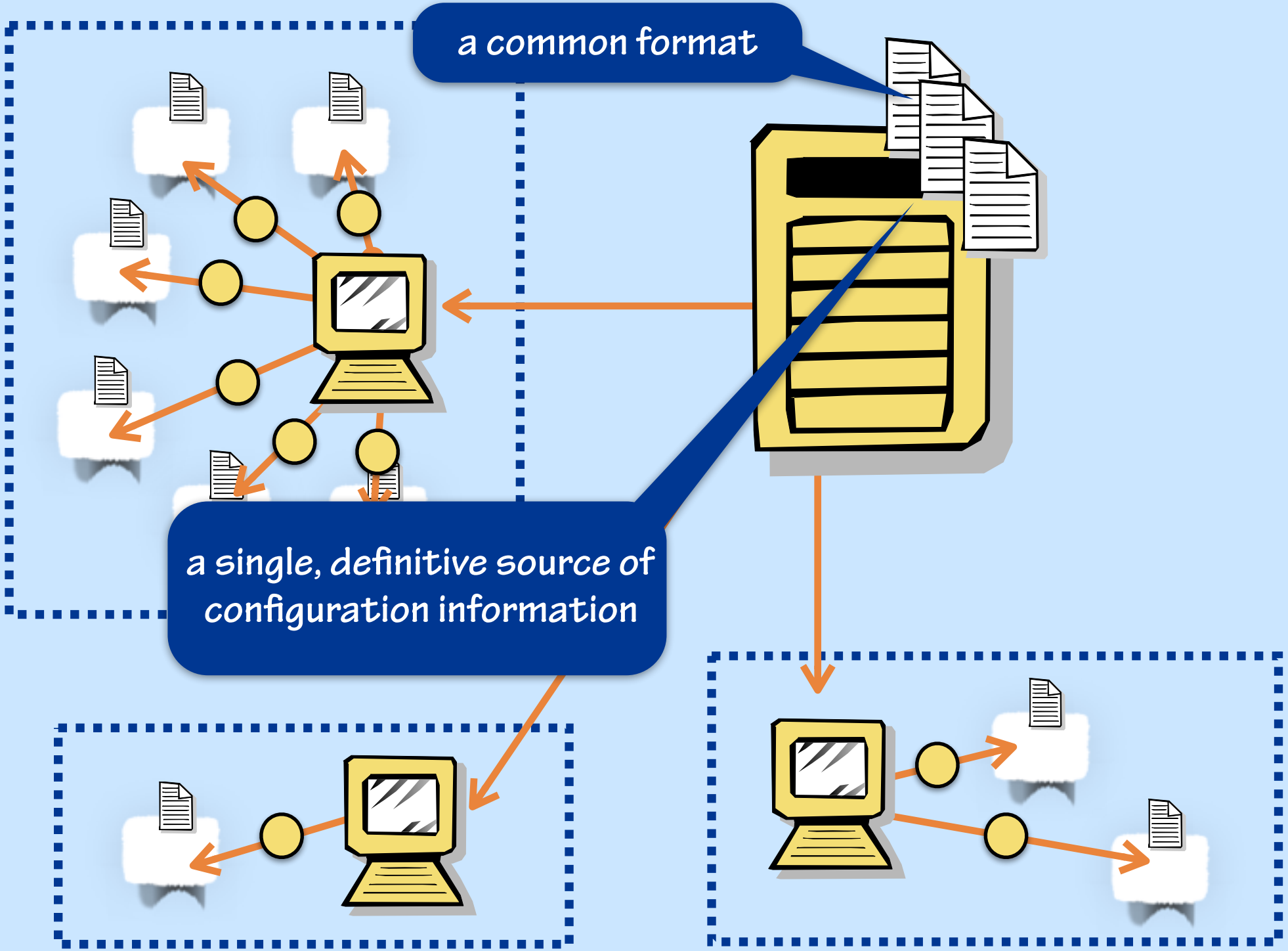
12.00-13.00 - Writing LCFG components (Paul)


web server
(apache)
application
(owncloud)
database
(mariadb)


web server
(apache)
application
(owncloud)
database
(mariadb)
ssh
package
manager
logrotate
firewall
(ufw)
backups


web server
(apache)
application
(owncloud)
database
(mariadb)
ssh
all in different formats!!!
package
manager
logrotate
firewall
(ufw)
backups


“client”
“profiles”
“server”
“component”


a common format
a single, definitive source of configuration information

Declarative Configuration

Configuration is data, not code

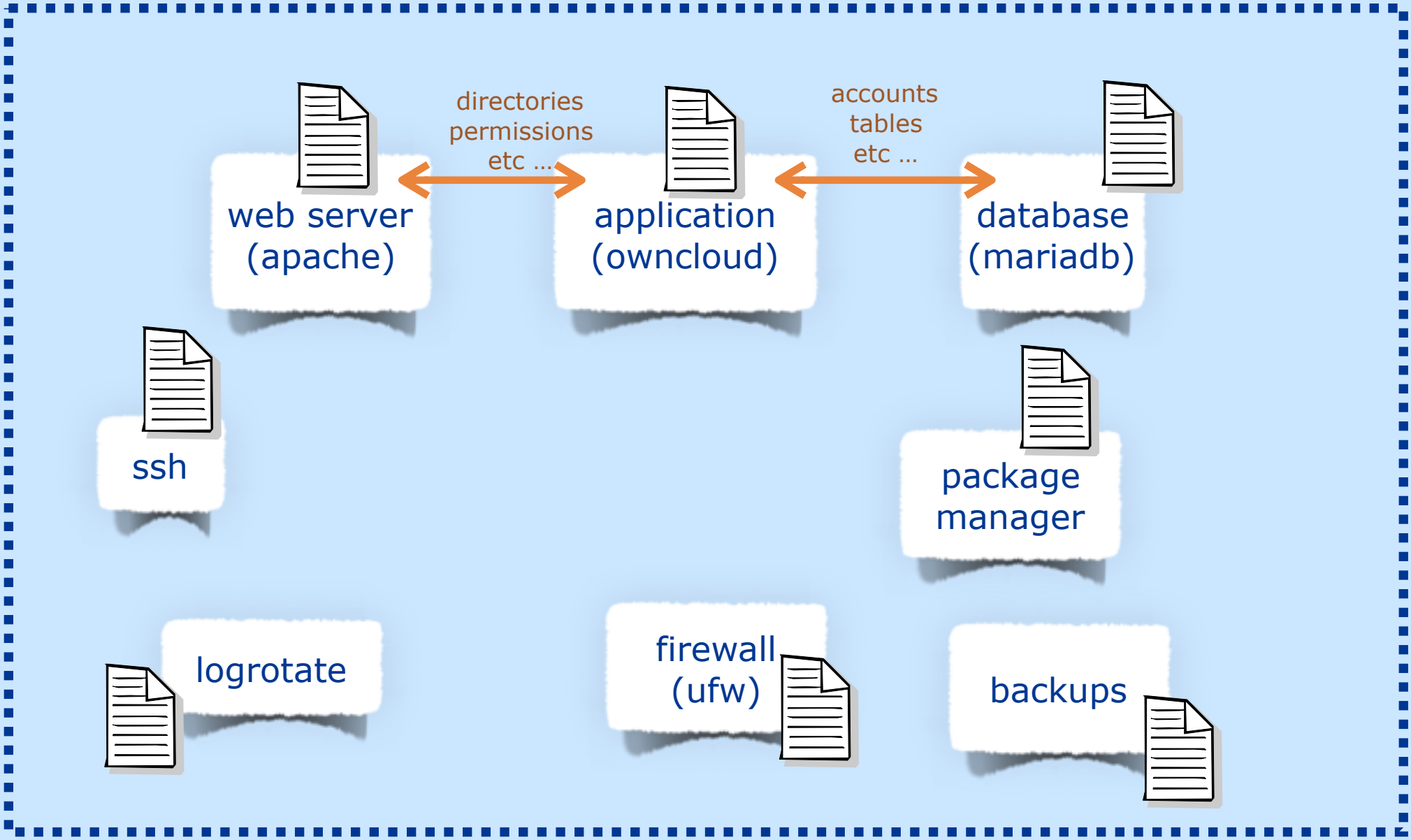
- the LCFG profiles are in XML
- we can generate the profiles in different ways
  - the existing LCFG language is very old (1994!)
  - L3 is a new, experimental language
- we can consume the profiles in different ways
  - to configure a machine
  - to do a complete security analysis of the whole site

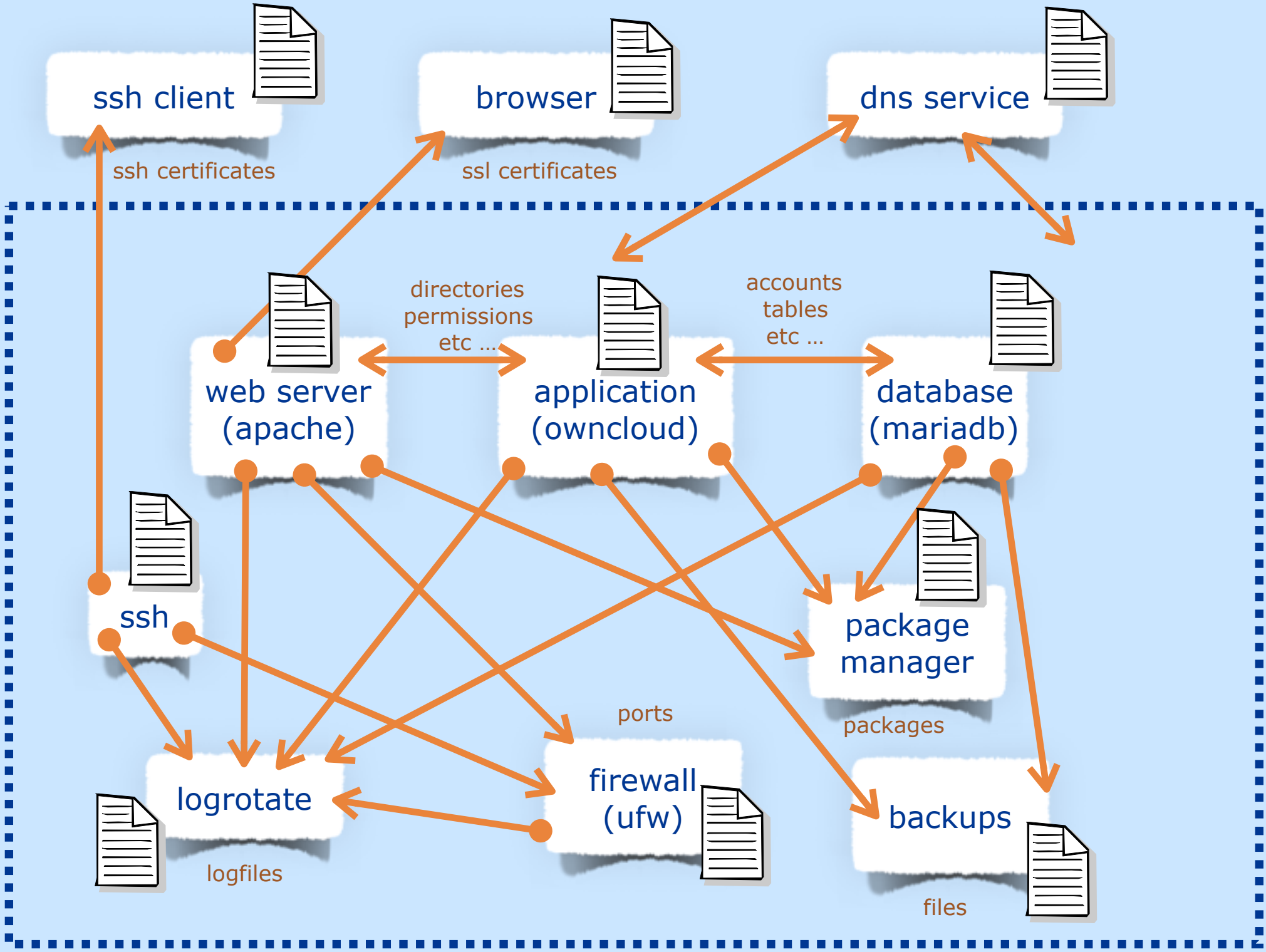
Deployment is separate from specification

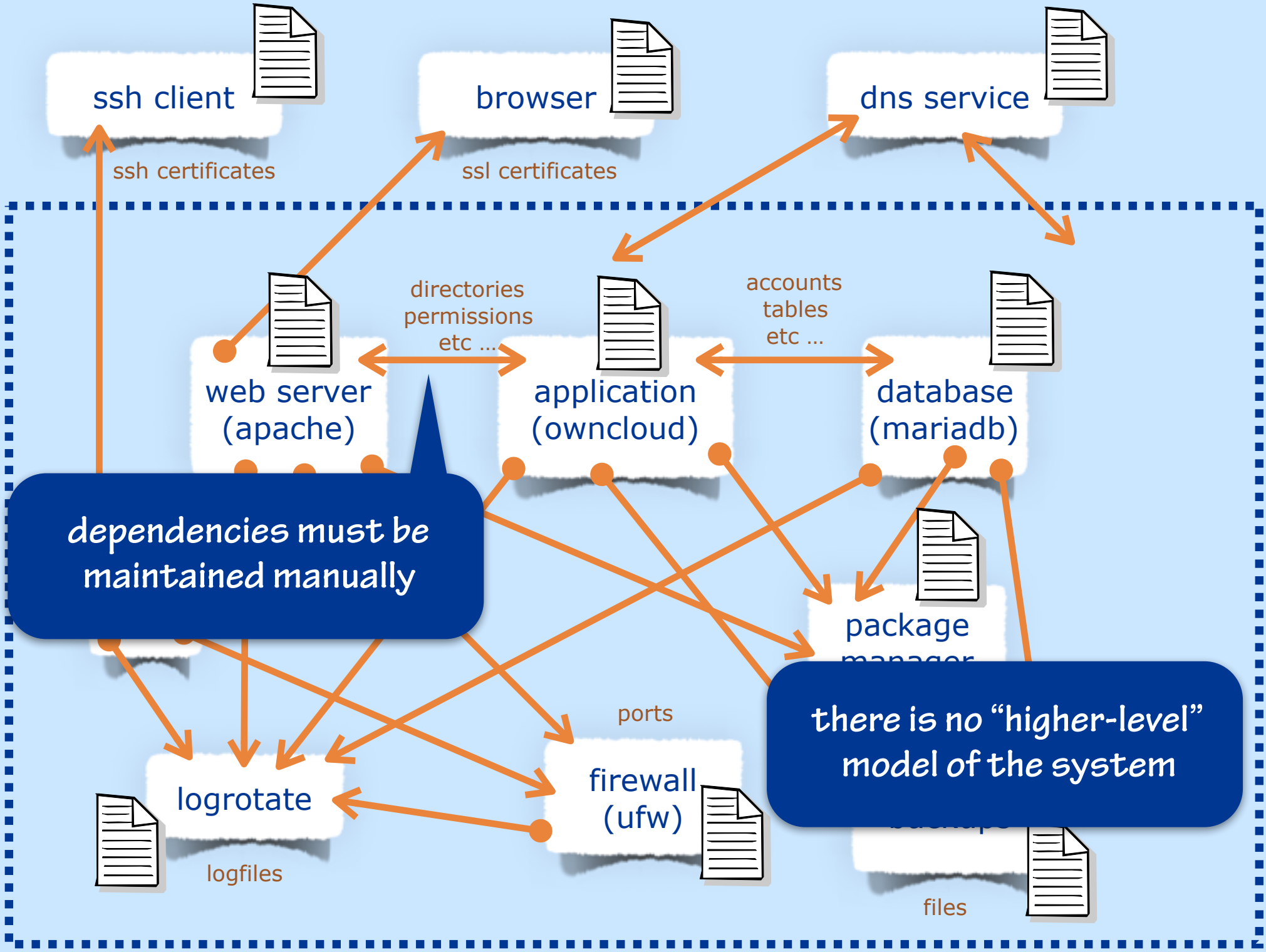
- components translate from the profile to system-specific formats
- we can use automated planning to control the deployment order

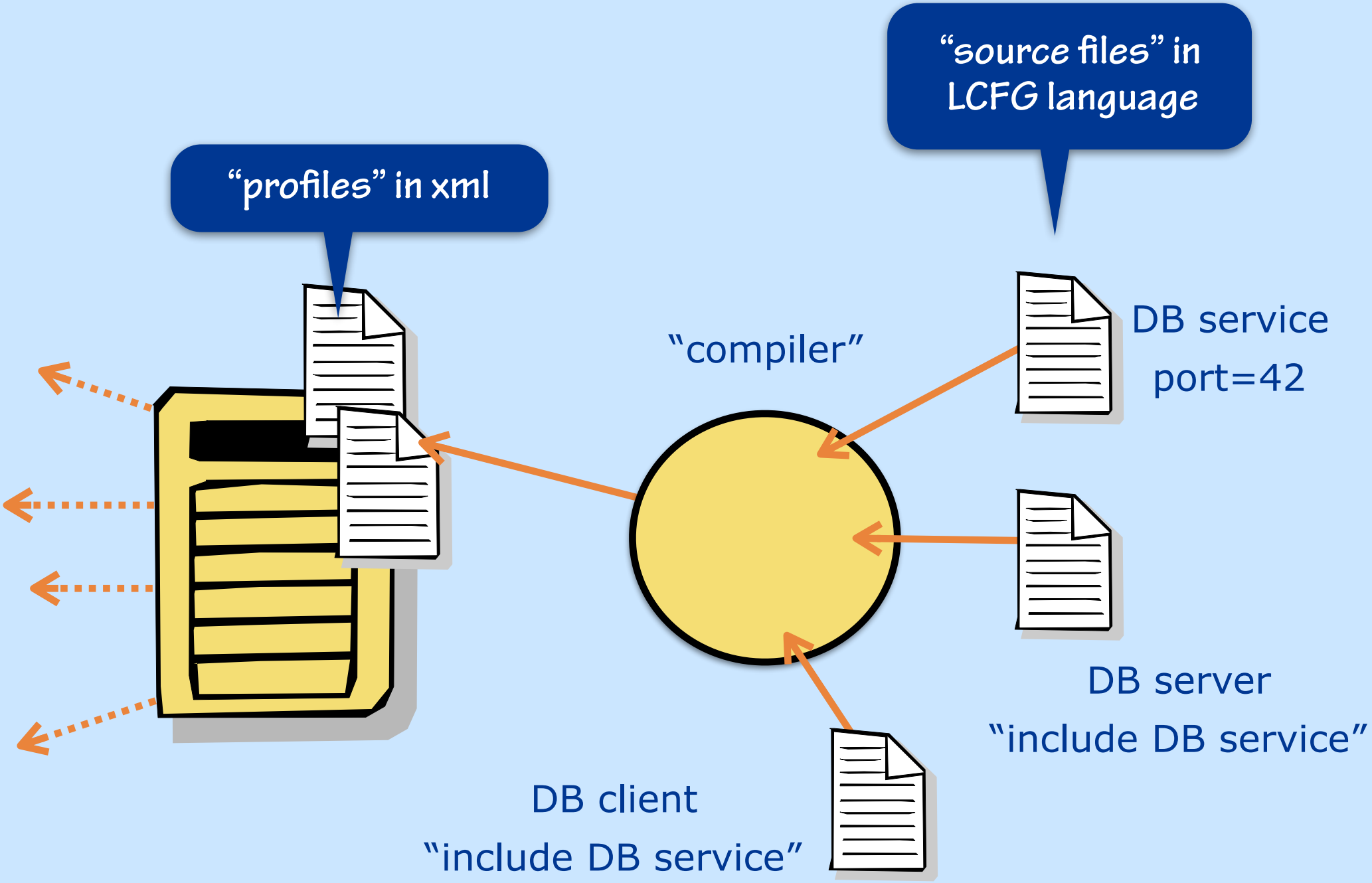
Many other systems are not “declarative”

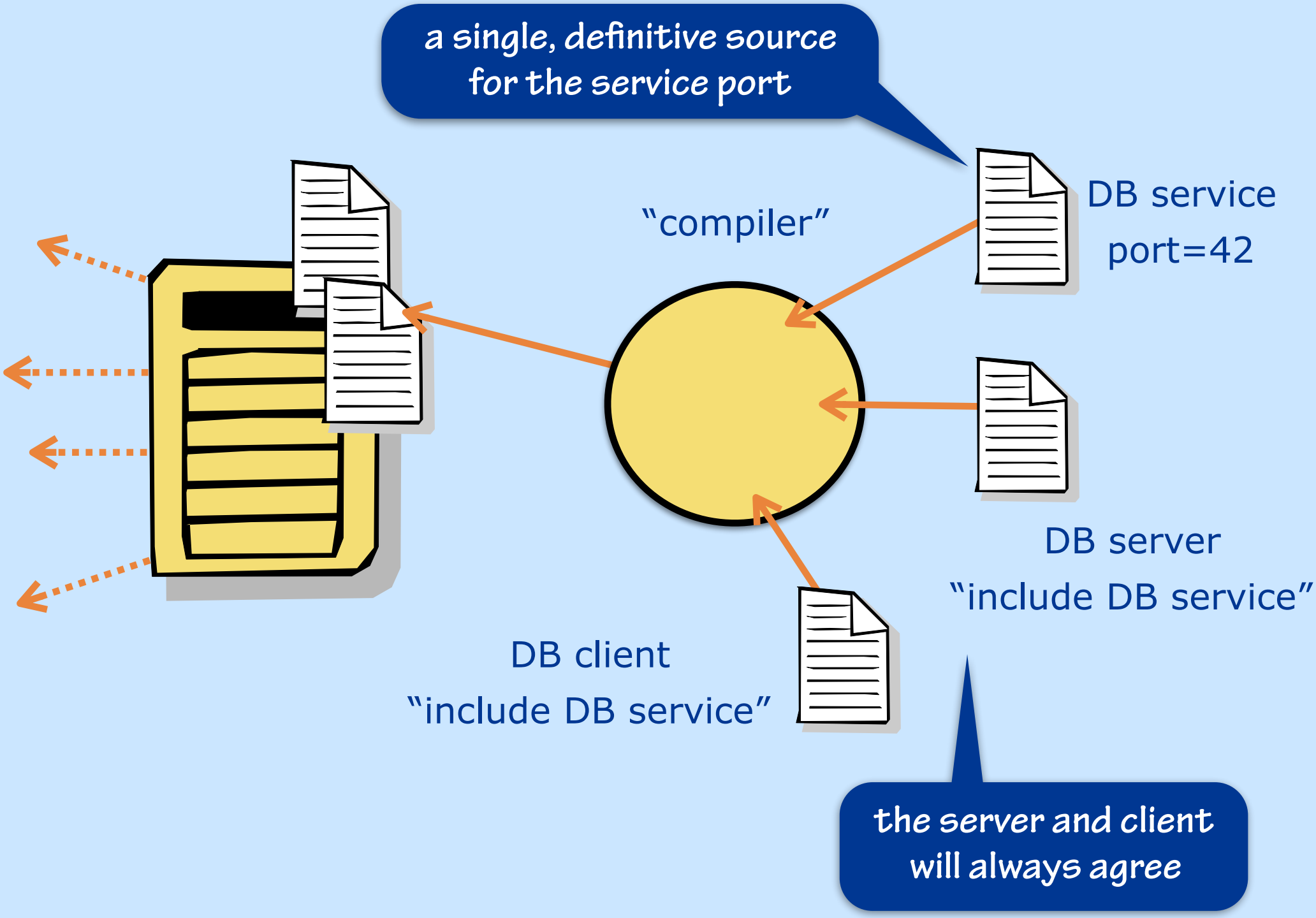
- they specify “how” (scripts), but not “what”
- having no explicit description of the desired configuration is a big disadvantage

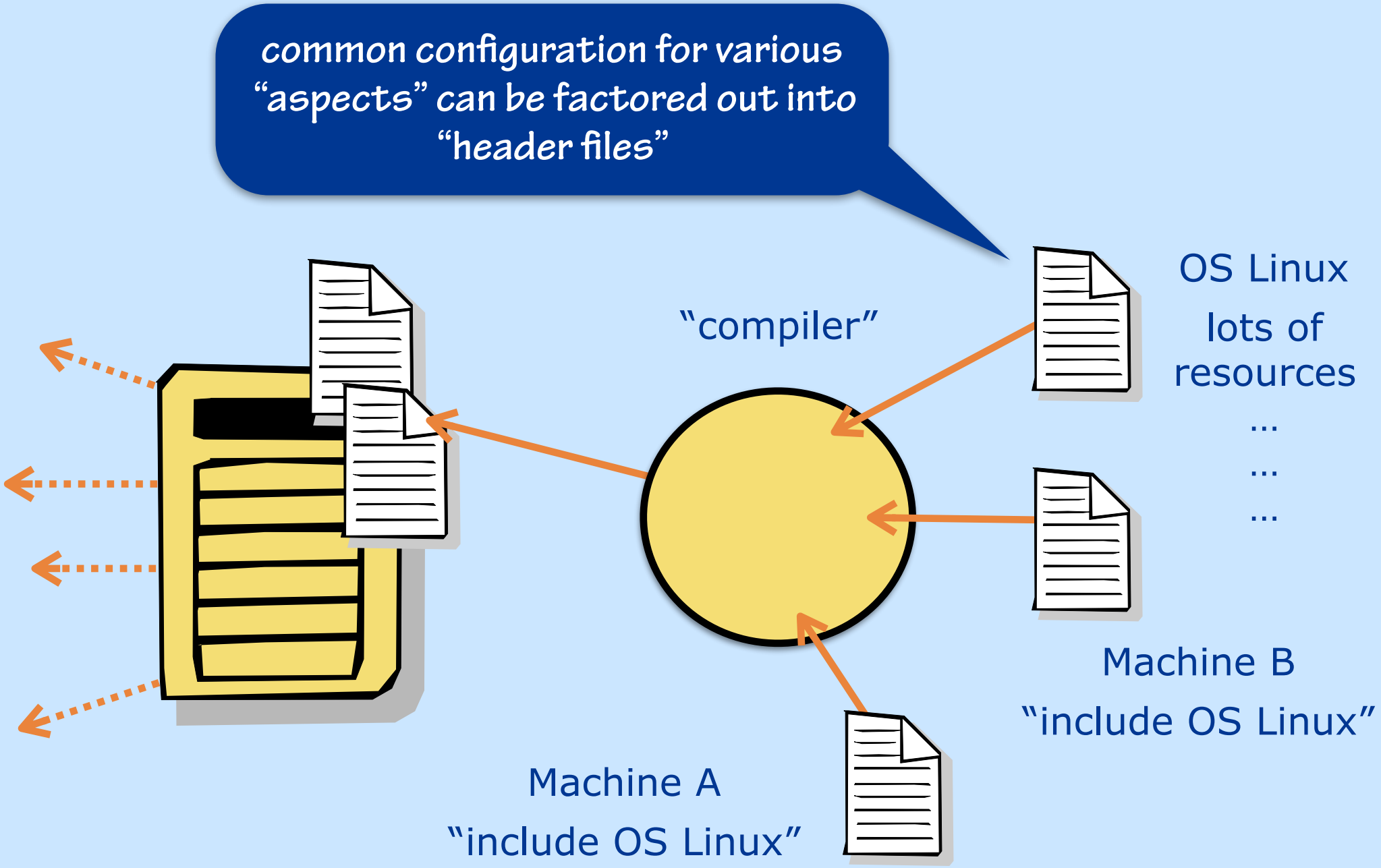

directories
permissions
etc ...
accounts
tables
etc ...
web server
(apache)
application
(owncloud)
database
(mariadb)
ssh
package
manager
logrotate
firewall
(ufw)
backups

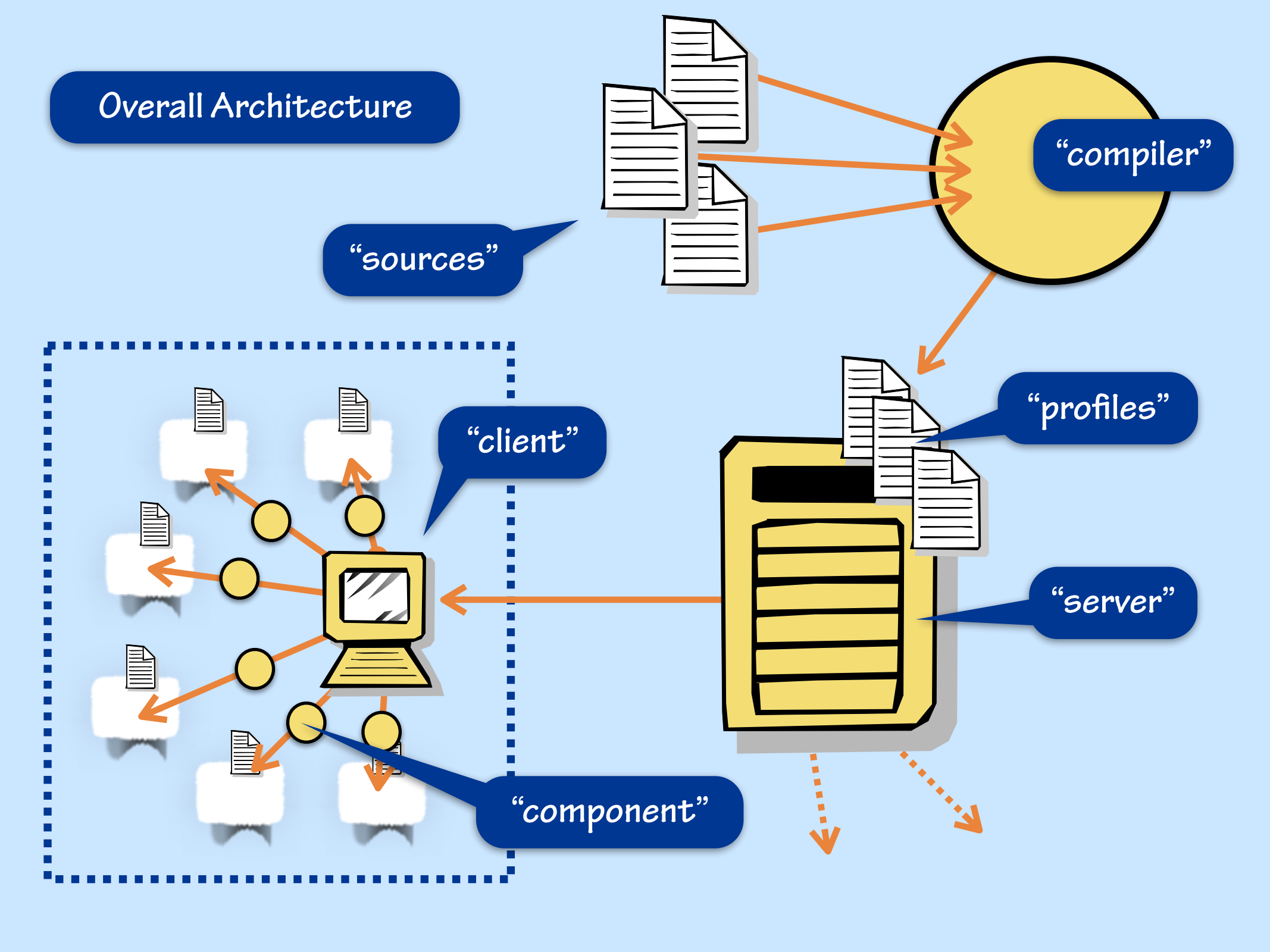

ssh client
browser
dns service
ssh certificates
ssl certificates
directories
permissions
etc ...
accounts
tables
etc ...
web server
(apache)
application
(owncloud)
database
(mariadb)
ssh
package
manager
packages
ports
logrotate
logfiles
firewall
(ufw)
backups
files


ssh client
browser
dns service
ssh certificates
ssl certificates
directories
permissions
etc ...
accounts
tables
etc ...
web server
(apache)
application
(owncloud)
database
(mariadb)
dependencies must be
maintained manually
package
manager
ports
there is no "higher-level"
model of the system
logrotate
firewall
(ufw)
logfiles
files


"source files" in LCFG language
"profiles" in xml
"compiler"
DB service
port=42
DB server
"include DB service"
DB client
"include DB service"


a single, definitive source for the service port
DB service port=42
"compiler"
DB server "include DB service"
DB client "include DB service"
the server and client will always agree


common configuration for various "aspects" can be factored out into "header files"
"compiler"
OS Linux
lots of resources
...
...
...
Machine B
"include OS Linux"
Machine A
"include OS Linux"


Overall Architecture
"compiler"
"sources"
"client"
"profiles"
"server"
"component"

What next?

Sessions today

- using the LCFG language to configure some existing components
- writing (or at least modifying) components

Resources

- the LCFG booklet is available online
  please do not distribute outside of Informatics
- the VM is available online
- the code is open source: www.lcfg.org

Research

- I'm interested in …
  - language semantics (e.g. aspect composition & conflict resolution), usability, deployment & planning, …
- Project opportunities …
  - Phd, MSc, INF4, INF3 research, summer projects …