CFEngine 3.x
Large-scale configuration mgmt.

P. Christeas, 2014
Ημέρες Συνεργασίας Ε/ΕΛ-ΛΑΚ
ΕΕΝ Hellas
How, Why?

My needs:
- Increasing number of machines
- Pro-active monitoring
- Parallel deployment
- Repetitive administration

CFEngine design:
- Scalability
- Promise theory
- Anomaly detection
- Change management
- Heterogenous adm.

IRC :)
CFEngine (academic) concepts

- Mark Burgess, creator, original author, prof. of Network and System Administration
- scientific approach, 90's
- Promise theory (declarative lang. of states)
- Change Management
- References, today's usage
Community vs. Enterprise

Community edition:
- Core components: cf-agent, server, monitord
- Design Center (console-only)

Enterprise edition:
- Knowledge management
- Web-interface
- Statistics, Graphs, feedback
Core components

- Written in C, standalone binaries (as root)
  - cf-agent
  - cf-serverd
  - cf-monitor
  - cf-execd, cf-key, cf-runagent, cf-promises
- Masterfiles, input files
- /var/cfengine/...
- git (down to policy server)
Policy flow

Policy Definition Point
/any/path

Distribution servers
/var/cfengine/masterfiles

End hosts
/var/cfengine/inputs
Promise theory, syntax

- Promises, declarative language
- Bundles (CFEngine 3)
- classes
- variables, actions (time-based, constraints)

bundle agent hello_world
{
  reports:
  any::
    "Hello World! I am $(sys.fqhost) and it is $(sys.date)"
  comment => "Prints a message, including hostname";
}
Agent promises

- files
  - copy
  - check
  - delete
  - perms
- commands
- methods
- processes
- packages
  - RPMs, DEBs etc.
- services
- storage
  (disk space, %)
- reports (logging)
- databases (?)
A file promise

files:
  Mageia::
  
  "/etc/ntp/step-tickers"
  handle=> "ntp_setup_tickers",
  copy_from=> remote_cp("$(def.dir_masterfiles)/Mageia/
                     ntp-step-tickers", "$(sys.policy_hub)")
  perms=> m("go+r")
  action=>if_elapsed_day;
Result?

- Many (different) machines on “auto pilot”
- 4 different setups, desktop machines!
- Alerts + rich forensics
- Tamper-proof
- Provisioning
- Continuous convergence to peace-of-mind
Zoom-out

(enterprise edition only?)

- Virtual environments
- Testing (Jenkins)
- Deployment, Risk management
- Multiple “branch” configurations