Topics in Configuration Management
BoFs @ Lisa ’10

Nico Schottelius, ETH Zurich
Hal Snyder and Jason Olson, Orbitz

10. November 2010
Introduction

Current situation

Nico’s stunning config world

Example implementation

Future
This BoFs

- This is **not** a vendor BoF
- Originally originated from a discussion
- CM is in: see IRC, meetings and ask yourself why you’re here
- Current situation of CM-Tools
- Brainstorming for a better world
This BoFs

- This is not a vendor BoF
- Originally originated from a discussion
  - CM is in: see IRC, meetings and ask yourself why you’re here
  - Current situation of CM-Tools
  - Brainstorming for a better world
This BoFs

- This is not a vendor BoF
- Originally originated from a discussion
- CM is in: see IRC, meetings and ask yourself why you’re here
- Current situation of CM-Tools
- Brainstorming for a better world
This BoFs

- This is not a vendor BoF
- Originally originated from a discussion
- CM is in: see IRC, meetings and ask yourself why you’re here
- Current situation of CM-Tools
- Brainstorming for a better world
This BoFs

- This is not a vendor BoF
- Originally originated from a discussion
- CM is in: see IRC, meetings and ask yourself why you’re here
- Current situation of CM-Tools
- Brainstorming for a better world
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
- Tried cfengine
- Using puppet with another sysadmin at ETH Zurich
- Seen many bugs and problems
- Thinks about writing a new CM
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
  - Tried cfengine
  - Using puppet with another sysadmin at ETH Zurich
  - Seen many bugs and problems
- Thinks about writing a new CM
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
- Tried cfengine
  - Using puppet with another sysadmin at ETH Zurich
  - Seen many bugs and problems
  - Thinks about writing a new CM
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
- Tried cfengine
- Using puppet with another sysadmin at ETH Zurich
  - Seen many bugs and problems
  - Thinks about writing a new CM
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
- Tried cfengine
- Using puppet with another sysadmin at ETH Zurich
- Seen many bugs and problems
- Thinks about writing a new CM
This guy

- Tries to optimise his (sysadmin) world
- Hates ugly design in software
- Tried cfengine
- Using puppet with another sysadmin at ETH Zurich
- Seen many bugs and problems
- Thinks about writing a new CM
CM tools

- Chef
- Puppet
- Cfengine
- Brcfg
- You name it
CM tools

- Chef
- Puppet
- Cfengine
- Brcfg
- You name it
CM tools

- Chef
- Puppet
- Cfengine
- Brcfg
- You name it
CM tools

- Chef
- Puppet
- Cfengine
- Brcfg
- You name it
CM tools

- Chef
- Puppet
- Cfengine
- Brcfg
- You name it
Facing the truth

- Bugs and design issues
  - Weired designs including reinvention of the wheel
  - Done parts right, breaking if heavily used
  - Wack documentation in many systems
  - You scream. The question is when.
Facing the truth

- Bugs and design issues
- Weired designs including reinvention of the wheel
  - Done parts right, breaking if heavily used
  - Wack documentation in many systems
  - You scream. The question is when.
Facing the truth

- Bugs and design issues
- Weired designs including reinvention of the wheel
- Done parts right, breaking if heavily used
  - Wack documentation in many systems
- You scream. The question is when.
Facing the truth

- Bugs and design issues
- Weired designs including reinvention of the wheel
- Done parts right, breaking if heavily used
- Wack documentation in many systems
- You scream. The question is when.
Facing the truth

- Bugs and design issues
- Weired designs including reinvention of the wheel
- Done parts right, breaking if heavily used
- Wack documentation in many systems
- You scream. The question is when.
Warning

- I know some config tools
- Thus I’m biased towards new systems
- I’m lazy
- I love Unix
Warning

- I know some config tools
- Thus I’m biased towards new systems
  - I’m lazy
  - I love Unix
Warning

- I know some config tools
- Thus I’m biased towards new systems
- I’m lazy
- I love Unix
I know some config tools
Thus I’m biased towards new systems
I’m lazy
I love Unix
Let’s go!

- Different try - turn the world around
  - Ignore any existing implementation
  - Though, not the concepts
  - Create lovely design of what I want
Let’s go!

- Different try - turn the world around
- Ignore any existing implementation
- Though, not the concepts
- Create lovely design of what I want
Let’s go!

- Different try - turn the world around
- Ignore any existing implementation
- Though, not the concepts
- Create lovely design of what I want
Let’s go!

- Different try - turn the world around
- Ignore any existing implementation
- Though, not the concepts
- Create lovely design of what I want
What do I want today?

- **Good** documentation
- UNIX: reuse of small apps
- Templates
- Discovery of client information (ssh keys, os type, etc.)
- Easy to write and/or already known language
What do I want today?

- **Good** documentation
- UNIX: reuse of small apps
  - Templates
  - Discovery of client information (ssh keys, os type, etc.)
  - Easy to write and/or already known language
What do I want today?

- Good documentation
- UNIX: reuse of small apps
- Templates
  - Discovery of client information (ssh keys, os type, etc.)
  - Easy to write and/or already known language
What do I want today?

- **Good** documentation
- UNIX: reuse of small apps
- Templates
- Discovery of client information (ssh keys, os type, etc.)
- Easy to write and/or already known language
What do I want today?

- **Good** documentation
- UNIX: reuse of small apps
- Templates
- Discovery of client information (ssh keys, os type, etc.)
- Easy to write and/or already known language
What do I want today? (2)

- Abstraction to dirty stuff (package managers, users, etc.)
- Library support
- Scaling large (> 10k clients shouldn’t be a real issue)
- Easy support for multiple servers
- Push and Pull architecture
What do I want today? (2)

- Abstraction to dirty stuff (package managers, users, etc.)
- Library support
  - Scaling large (> 10k clients shouldn’t be a real issue)
  - Easy support for multiple servers
  - Push and Pull architecture
What do I want today? (2)

- Abstraction to dirty stuff (package managers, users, etc.)
- Library support
- Scaling large (> 10k clients shouldn’t be a real issue)
- Easy support for multiple servers
- Push and Pull architecture
What do I want today? (2)

- Abstraction to dirty stuff (package managers, users, etc.)
- Library support
- Scaling large (> 10k clients shouldn’t be a real issue)
- Easy support for multiple servers
- Push and Pull architecture
What do I want today? (2)

- Abstraction to dirty stuff (package managers, users, etc.)
- Library support
- Scaling large (> 10k clients shouldn’t be a real issue)
- Easy support for multiple servers
- Push and Pull architecture
Nice to have

- (Almost) No requirements on the clients
- Support Windows\^Wnon-Unix-Variants
Nice to have

- (Almost) No requirements on the clients
- Support Windows\Wnon-Unix-Variants
I can reinvent the wheel

But cycling alone is boring

Your input here
How would I...?

- **Good documentation**: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- Good documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- Good documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...?

- **Good** documentation: manpages sections 1 and 7
- **UNIX**: reuse of small apps: use what’s available
- **Templates**: I don’t. You do. man stdout.
- **Discovery of client information**: ssh + see above
- **Easy to write and/or already known language**: I’m dumb. Use shell.
- **Abstraction to dirty stuff**: Define types
- **Library support**: Weired ideas existing
- **Scaling large**: As easy as adding a new server
- **Easy support for multiple servers**: Use a VCS
- **Push and Pull architecture**: Easy to do, use triggers.
How would I...

- **Good** documentation: manpages sections 1 and 7
- UNIX: reuse of small apps: use what’s available
- Templates: I don’t. You do. man stdout.
- Discovery of client information: ssh + see above
- Easy to write and/or already known language: I’m dumb. Use shell.
- Abstraction to dirty stuff: Define types
- Library support: Weired ideas existing
- Scaling large: As easy as adding a new server
- Easy support for multiple servers: Use a VCS
- Push and Pull architecture: Easy to do, use triggers.
How would I...

- (Almost) No requirements on the clients: Shell + Unix
- Support Windows\non-Unix-Variants: Cygwin
- Other stuff: You.
How would I...?

- (Almost) No requirements on the clients: Shell + Unix
- Support Windows\non-Unix-Variants: Cygwin
- Other stuff: You.
How would I...?

- (Almost) No requirements on the clients: Shell + Unix
- Support Windows\non-Unix-Variants: Cygwin
- Other stuff: You.
What’s next?

- I imagine some people are working on this.
- I assume if there was a release, you would find it.
What’s next?

- I imagine some people are working on this
- I assume if there was a release, you would find it.
Thanks for listening!

More stuff can be found on http://www.nico.schottelius.org/blog/