A Continuously Updated CMDB
using
The Assimilation Project

LCA 2014

#AssimProj @OSSAlanR

http://assimproj.org/

http://bit.ly/LCA2014-SysAdmin

Alan Robertson <alanr@unix.sh>

Assimilation Systems Limited

http://assimilationsystems.com

Assimilation Project Scope

Zero-network-footprint continuous **Discovery**

Integrated with extreme-scale **Monitoring**

=> Discovery creates a graph-based **CMDB**

Using a CMDB for Risk Management/Mitigation

- Intrusions
- Licensed Software
- Audit Risk
- System modeling
- Outages

linux.conf.au
06 January
2014

Why a Configuration Management Database (CMDB)?

- **Documentation**: incomplete, incorrect
- **Dependencies**: unknown
- **Planning**: Needs accurate data
- **Best Practices**: Verification needs data
- **Compliance**
- Our Discovery: continuous, low-profile

Discovery Features

- Continuous Discovery
- Zero network footprint
- Discover dependency information
- Discovery drives monitoring
- Easily extensible
- Configuration-free setup (!)

What do we discover?

- IP and MAC addresses (servers, etc)
- Services and service details
- Switches, switch connections and settings
- Installed services
- OS configuration
- Whatever you want ;-)

Architectural Elements

- Collective Management Authority (CMA) – one per installation
- Nanoprobes (agents) – one per system

How does discovery work?

Nanoprobe scripts perform discovery

- Each discovers one kind of information
- Can take arguments from environment
- Output **JSON**

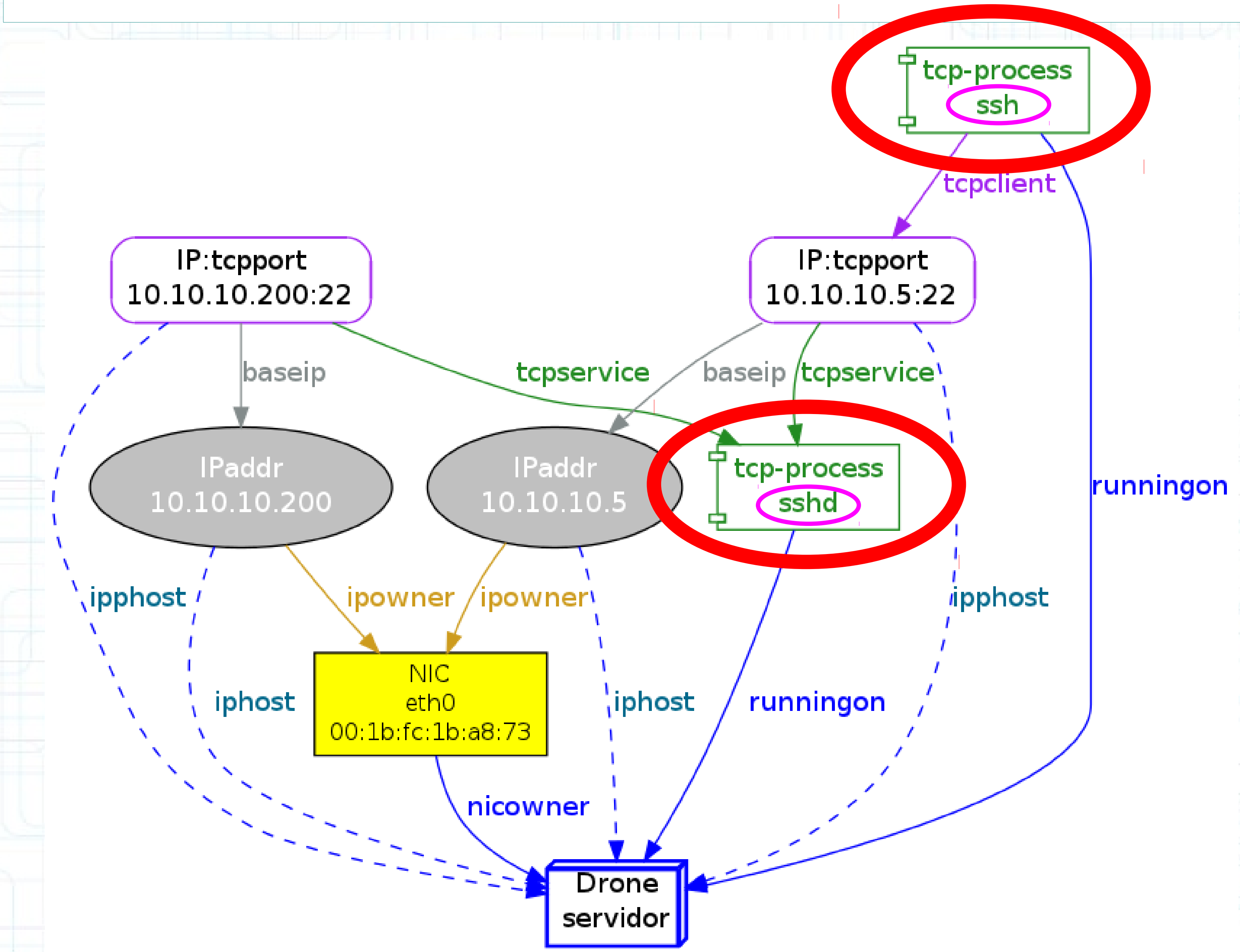
CMA stores Discovery Information

- JSON stored in Neo4j database
- CMA discovery plugins => graph nodes and relationships

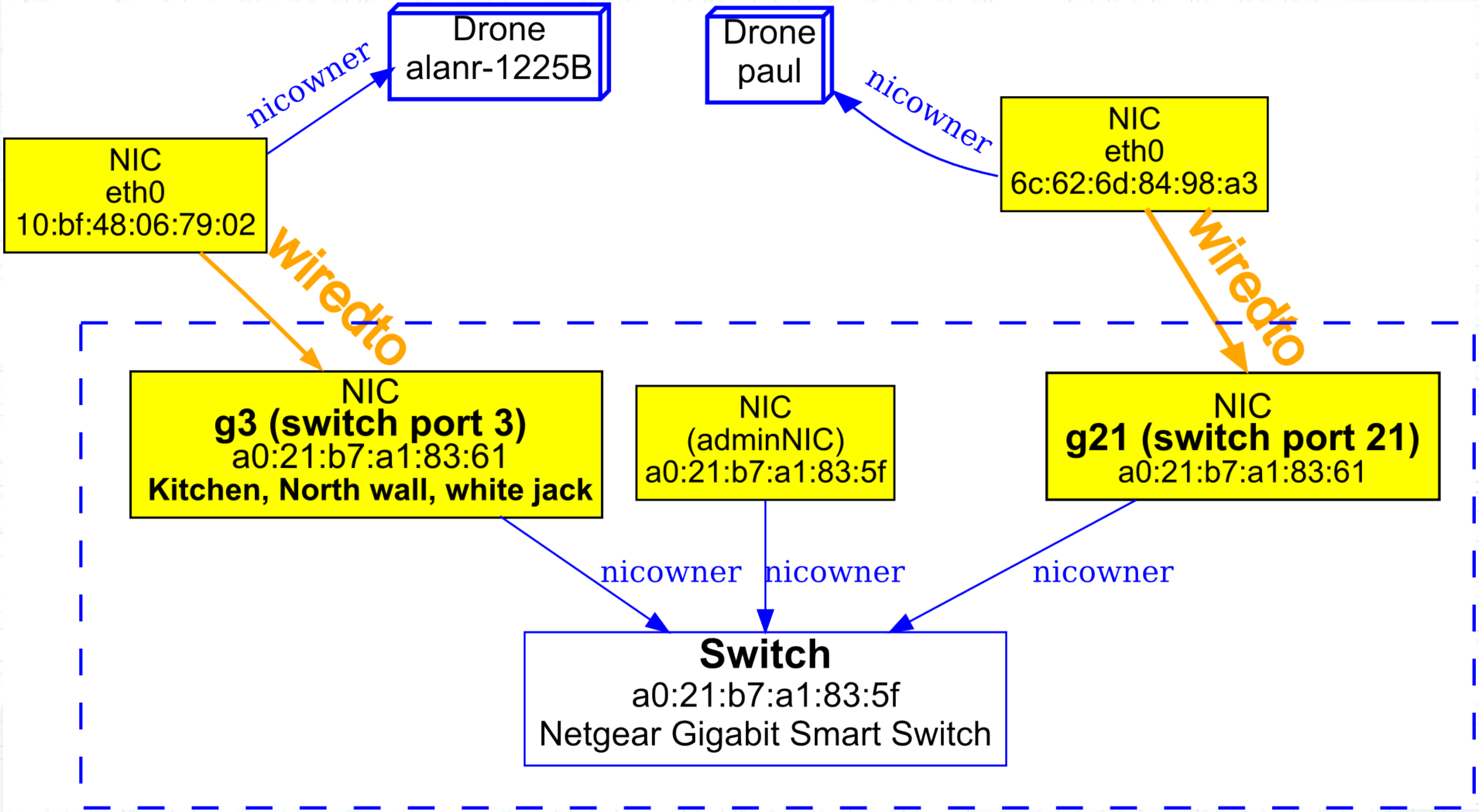
OS discovery JSON Snippet

```
{   "nodename":                 "alanr-1225B",
    "operating-system":         "GNU/Linux",
    "machine":                  "x86_64",
    "processor":                "x86_64",
    "hardware-platform":        "x86_64",
    "kernel-name":              "Linux",
    "kernel-release":           "3.8.0-31-generic",
    "kernel-version":           "#46-Ubuntu SMP ...",
    "Distributor ID":           "Ubuntu",
    "Description":              "Ubuntu 13.04",
    "Release":                  "13.04",
    "Codename":                 "raring"
}
```

ssh -> sshd dependency graph



Switch Discovery Data from LLDP (or CDP)



CRM transforms LLDP (CDP) Data to JSON

Current Status

- First release April 2013
- Great unit tests
- Nanoprobe code works well
- Several discovery methods written
- Discovery => Automatic Monitoring (WOOT!)
- UI development underway
- Licensed under GPL: commercial options available

Get Involved!

We need every talent!

- **Early adopters (SysAdmins(!))**
- Testers
- Designers
- Developers (C,Python, Shell, PowerShell, JavaScript)
- Porters (esp Windows)
- Promoters, publicists
- Packagers
- And so on...

Resistance Is Futile!

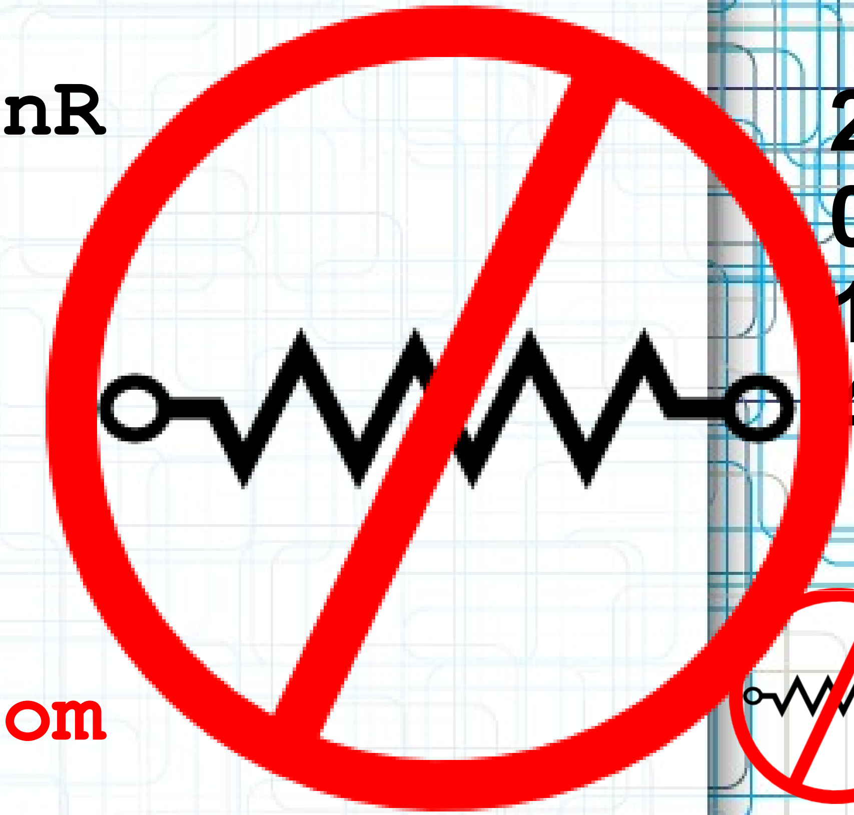
Mailing List

bit.ly/AssimML

#AssimProj @OSSAlanR

Project Web Site

assimproj.org



Blog

techthoughts.typepad.com

assimilationsystems.com

Why a graph database? (Neo4j)

- Humans describe systems as graphs
- Dependency & Discovery information: graph
- Speed of graph traversals depends on size of subgraph, not total graph size
- Root cause queries ⇨ graph traversals – notoriously slow in relational databases
- Visualization is Natural
- Schema-less design: good for constantly changing heterogeneous environment
- Graph Model === Object Model

LCA 2014

linux.conf.au
06 January
2014

ssh *Client* JSON Snippet (from netstat and /proc)

```
"ssh": {
  "exe":                  "/usr/sbin/ssh",
  "cmdline":              [ "ssh", "servidor" ],
  "uid":                  "alanr",
  "gid":                  "alanr",
  "cwd":                  "/home/alanr/monitor/src",
  "clientaddrs": {
    "10.10.10.5:22": {
      "proto":            "tcp",
      "addr":             "10.10.10.5",
      "port":             22
    },          and so on...
```

sshd *Service* JSON Snippet (from netstat and /proc)

```
"sshd": {
  "exe":                 "/usr/sbin/sshd",
  "cmdline":             [ "/usr/sbin/sshd", "-D" ],
  "uid":                 "root",
  "gid":                 "root",
  "cwd":                 "/",
  "listenaddrs": {
    "0.0.0.0:22": {
      "proto":           "tcp",
      "addr":            "0.0.0.0",
      "port":            22
    },
```

and so on...

L
C
A

2
0
1
4

A multi-dimensional demo

- Demonstrate basic capabilities
  - Discovery
  - Automatic monitoring configuration
  - Monitoring – failures / successes
- No configuration was supplied
  - *everything* comes from discovery