Open Telemetry and Model-Driven Configuration

Ken Wilson

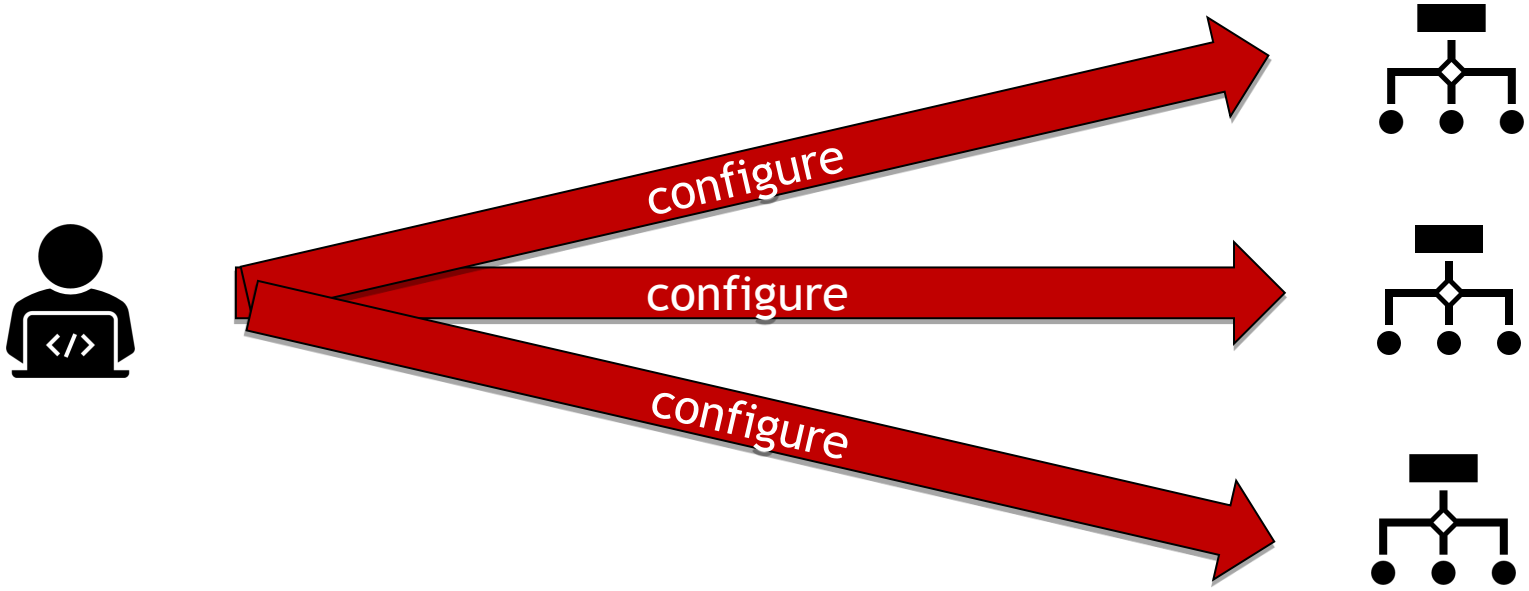
5-Sept-19

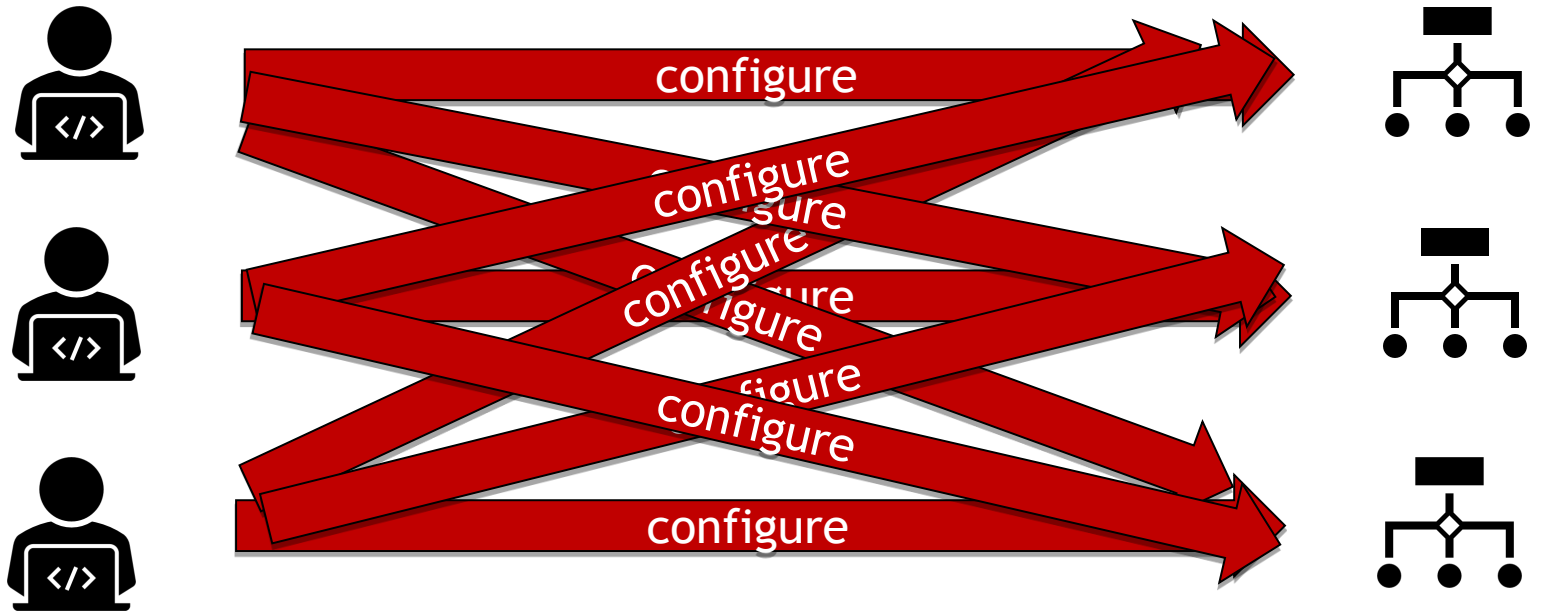
AusNOG 2019

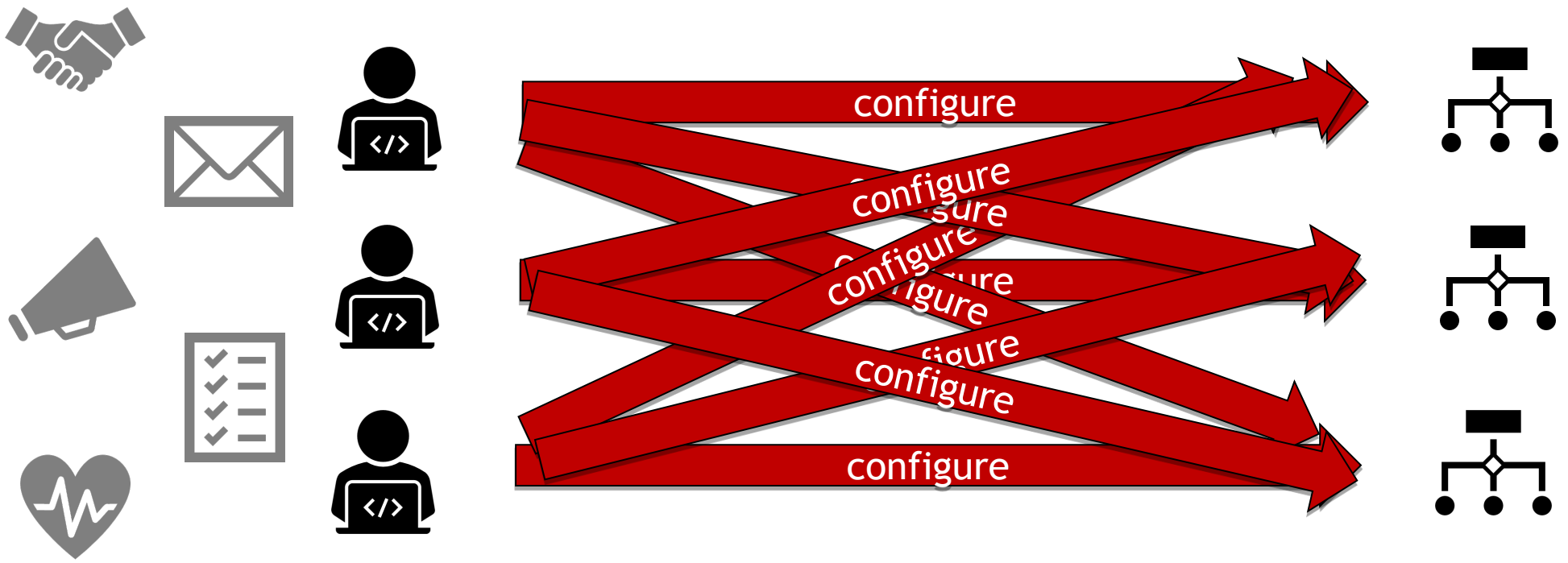
Overview

- Motivation
- Exciting news from the world of software development
- Strategies for automation
- Standards
- Further Reading

How it starts





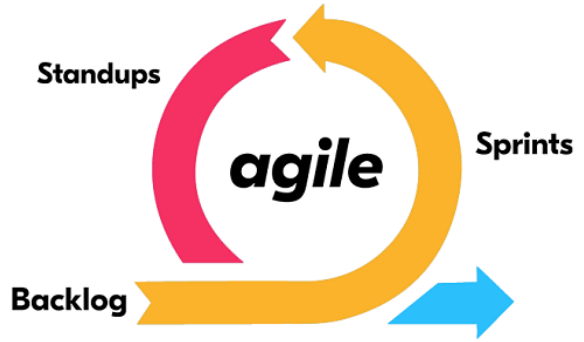


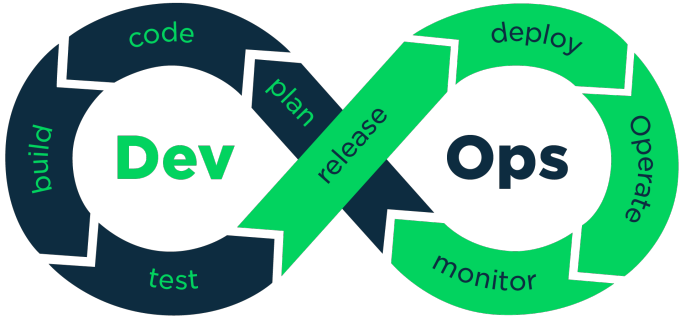
Trouble with configuring via the CLI

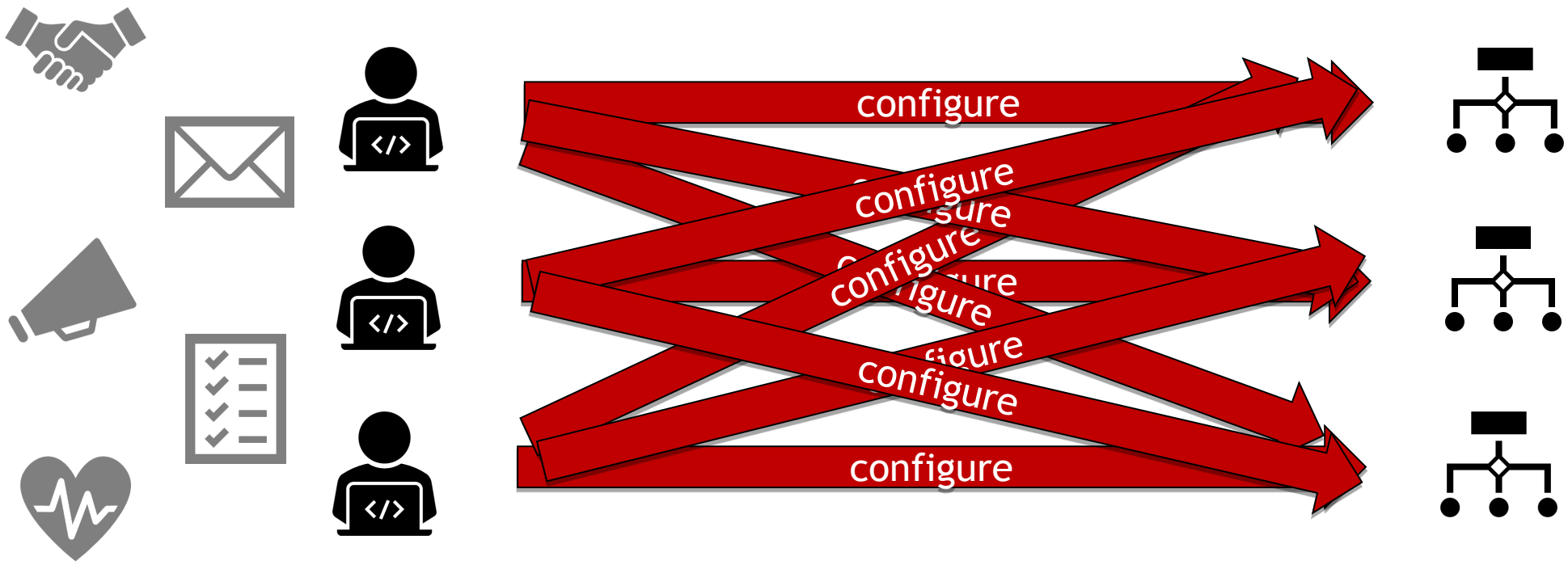
- Configs applied and maintained by hand
- Vendor-specific syntax, ever-changing
- Inconsistent configurations, unexplained special cases
- Configurations are forgotten; until... reactive break-fix model
- Workarounds: brute force, diligence, hard work, RANCID
- Fragile scrapers break on upgrades
- Human-oriented interfaces are for humans

Exciting news from the world of software

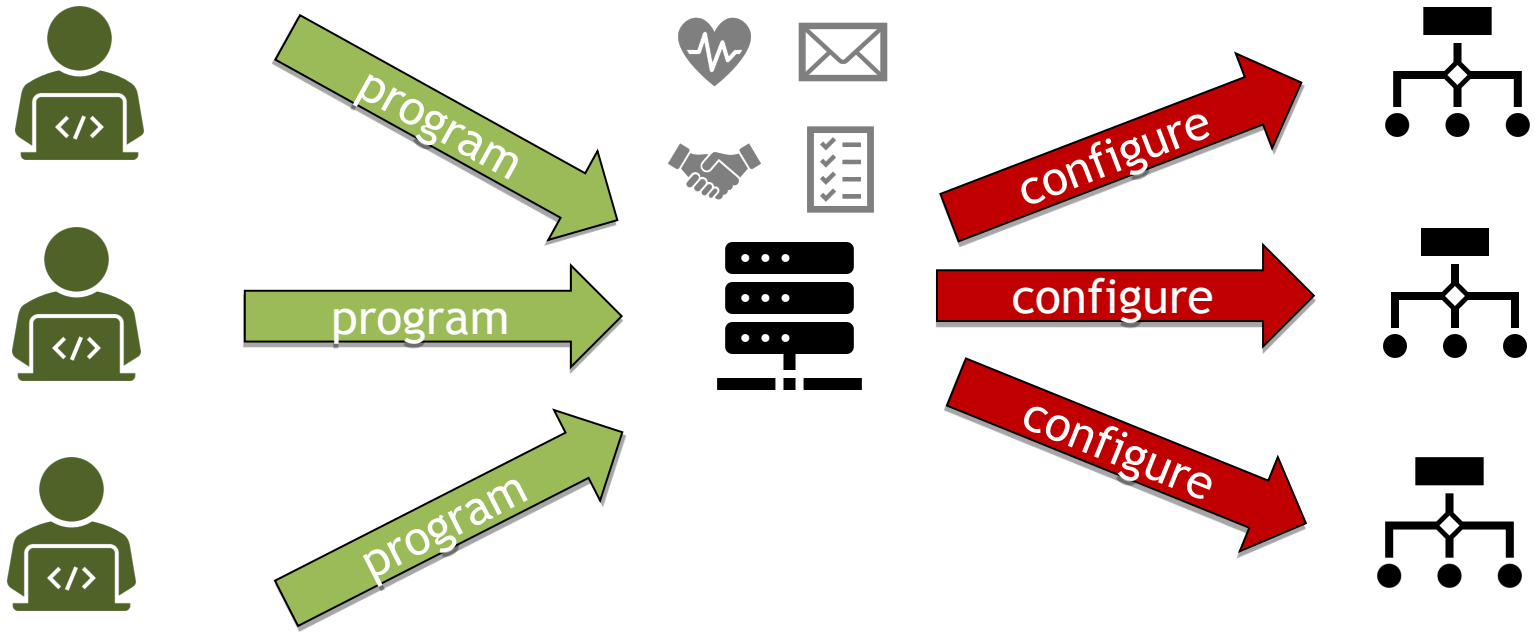
- Agile practices
  - Lighter but more intense teamwork
- git + pull requests
  - collaborative text editing, focused review
- CI/CD + tests + sandboxes
  - safety nets are safer
- DevOps
  - the best tools and practices adopted by operations



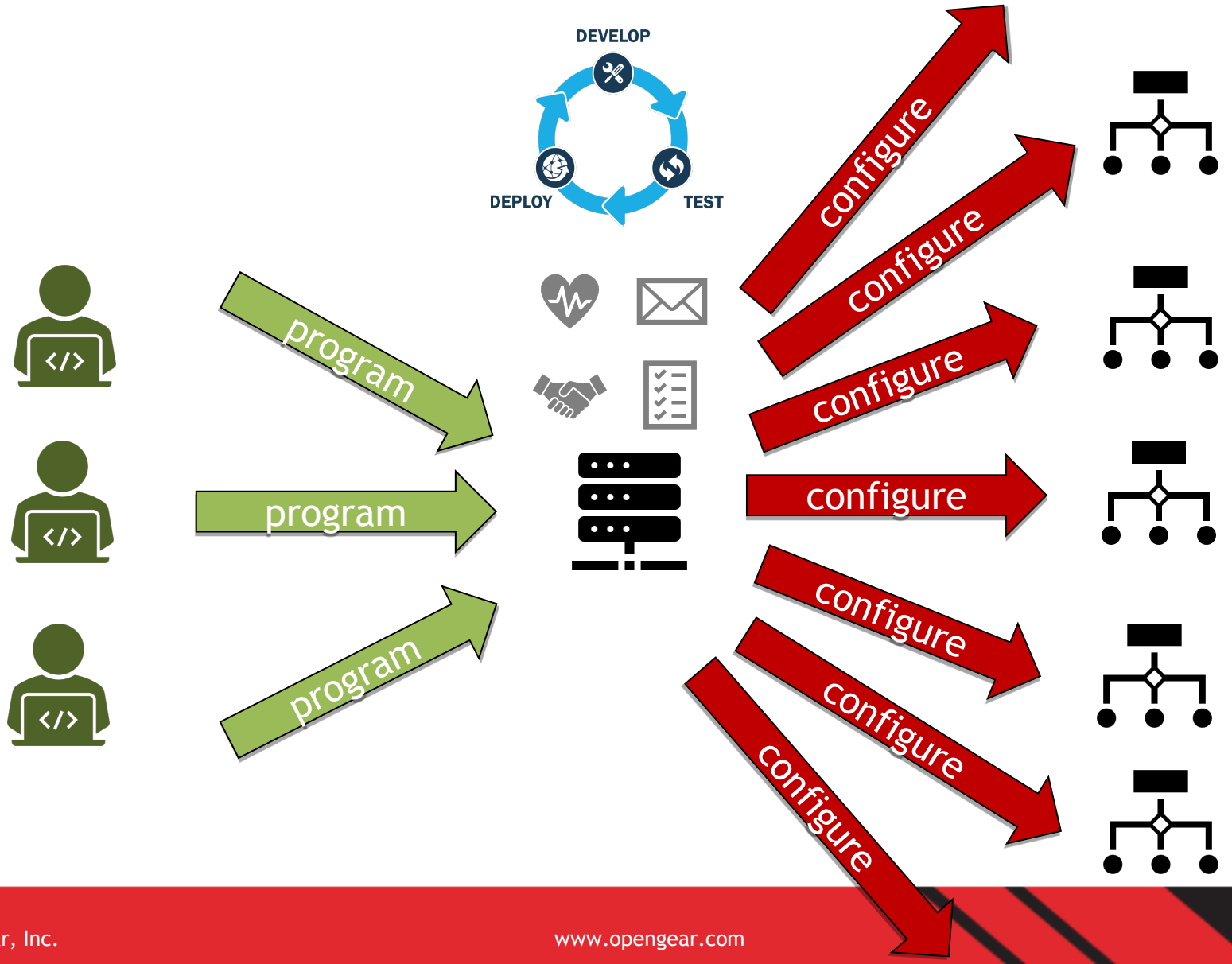




1:100



1:100



1:1000

Pre-conditions for automation

- **Inventory**: do you have good detail of all devices?
- **Requirements**: business's expectation of function level; now & plans
- **Standards**: which model/vocabulary to use in automation
- **Telemetry**: feedback channels
- **Automation**: control channels; tooling
- **Trust**: get experience, confidently predict automation's behaviour
- **other**: Budget, Stakeholders, Suppliers

	Provisioning	Monitoring	Security	…
Requirements	?	?	?	?
Automation	?	?	?	?
Telemetry	?	?	?	?
Inventory	?	?	?	?
Standards	?	?	?	?
Suppliers	?	?	?	?
Partners	?	?	?	?
Budget	?	?	?	?
Trust	?	?	?	?

Traditional Orchestration Systems

- Orchestration is not a new thing in the compute world.
- Multiple Solutions:
  - Ansible
  - Salt
  - Puppet
  - Chef
- These tools can be used for Network Orchestration, but generally still use the vendor CLI underneath.

Traditional Orchestration Systems (cont'd)

- Telemetry is also something that isn't well handled by traditional orchestration systems.
  - Primarily concerned with system state, not system performance
  - Often the configuration of telemetry is managed
    - SNMP
    - Netflow
  - Telemetry occurs via a different channel/service

Standards

IETF NETCONF

- 2006 RPC-based protocol for configuring network devices
- "SNMP done right"
- Replaces CLI-based programmatic interfaces (perl/expect over SSH)
- Installs, manipulates configuration
- Can validate config before activation
- Atomic commit/transaction *across multiple devices*
- Structured message and errors (XML/JSON)
- Includes streaming telemetry, with endpoints defined in models

NETCONF landscape

- NETCONF – the client-server protocol
  - NETCONF: sends XML over SSH
  - RESTCONF: sends XML or JSON over HTTP
- YANG – the modeling language for NETCONF
  - like MIBs' ASN.1 notation, uses YAML
  - Can describe the configuration parameters for an object, as well as telemetry endpoints.

NETCONF: XML over SSH

```
$ ssh -oHostKeyAlgorithms=+ssh-dss root@ios-xe-mgmt.cisco.com -p 10000 -s netconf

S: <hello> <capabilities> … </capabilities> </hello>
   ]]>]]>
C: <hello> <capabilities> … </capabilities> </hello>
```

NETCONF: XML over SSH (continued)

```
C: <rpc>
     <get-config>
       <source><running/></source>
     </get-config>
   </rpc>
S: <rpc-reply>
     <data>
       <interfaces><interface> <name>eth0</name> <enabled>true<… </interfaces>
     </data>
   </rpc-reply>
   ]]>]]>
```

NETCONF: XML over SSH - Telemetry

C:
```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
      <establish-subscription xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications”
                xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
                <stream>yp:yang-push</stream>
                <yp:xpath-filter> /cdp-ios-xe-oper:cdp-neighbour-details/cdp-neighbour-detail</yp:xpath-filter>
                <yp:dampening-period>0</yp:dampening-period>
      </establish-subscription>
</rpc>
```
S:
```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
      <subscription-result xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications" xmlns:notif
                bis="urn:ietf:params:xml:ns:yang:ietf-event-notifications">notif-bis:ok</subscription-result>
      <subscription-id xmlns="urn:ietf:params:xml:ns:yang:ietf-event-notifications">2147483685</
                subscription-id>
</rpc-reply>
```

NETCONF: XML over SSH – Telemetry (cont)

S:

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2017-05-09T21:34:51.74Z</eventTime>
<push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <subscription-id>2147483685</subscription-id>
    ....
</push-update>
      </notification>
```

YANG example

https://wikipedia.org/wiki/YANG#Example

```
module example-sports {

  namespace "http://example.com/example-sports";
  prefix sports;

  import ietf-yang-types { prefix yang; }

  typedef season {
    type string;
    description
      "The name of a sports season, including the type and the year,
e.g,'Champions League 2014/2015'.";
  }

  container sports {
    config true;

    list person {
      key name;
      leaf name { type string; }
      leaf birthday { type yang:date-and-time; mandatory true; }
    }
    list team {
      key name;
      leaf name { type string; }
      list player {
        key "name season";
        unique number;
        leaf name { type leafref { path "/sports/person/name"; }  }
        leaf season { type season; }
        leaf number { type uint16; mandatory true; }
        leaf scores { type uint16; default 0; }
      }
    }
  }
}
```

YANG Models

- Vendor Provided
  - Cisco/Juniper/Nokia and others all provide models for their configuration
  - Not compatible with each other
- IETF Netmod Models
  - Basic building blocks
  - Designed through the existing IETF working group process
  - This can be a slow process

OpenConfig

- OpenConfig
  - Specified by a working group of operators, rather than vendors
  - Focussed on capabilities required by operators
  - Faster frequency of updates in response to new needs.

Where to get YANG models

- yangcatalog.org
  - https://github.com/YangModels/yang (yangcatalog.org)
- openconfig.org
  - https://github.com/openconfig/public

Tooling

- YDK/NCClient
  - Cisco provided YANG Development Kit
  - Designed for programmatic access and configuration via NETCONF/RESTCONF/gNMI
  - https://archive.nanog.org/sites/default/files/20161017_Alvarez_Ok_We_Got_v1.pdf
- Salt/Ansible/Puppet
  - NETCONF support integrated into a number of these orchestration tools
    - This is where I'd start
- OpenDaylight
  - SDN controller that speaks NETCONF/RESTCONF both for configuring it, and downstream devices
  - Generally seen in carrier networks, rather than enterprise