TUNING THE WARP DRIVE WITH LAFORGE

A NEW TOOL FOR BUILDING SECURITY COMPETITIONS

BSIDESLV 2018

AGENDA

1. Introduction
2. The Problem
3. The Solution
4. The Future

THE CREW

Alex Levinson

Senior Security Engineer @ Uber

Speciality in tool dev & red teaming

CPTC Black Team Lead, NCCDC Red Team

Dan Borges

Senior Red Teamer

Wizard of detection exercises

CPTC OSINT Team Lead, NCCDC Red Team

SECURITY COMPETITIONS

Invest in the Future

They get *the next generation* excited about security.

Build Real World Skills

Classroom and labs only go so far, at some point you need to attack or defend.

Everyone Has Fun

You learn, you win, and you get to break and build things you may not usually.

COLLEGIATE PENTESTING COMPETITION (CPTC)

Competitor Consulting R' Us

Competitors must perform a penetration test of a fictitious company:

- Each team is a "consulting firm" performing a test.
- They are provided a full network to perform their testing.
- Volunteers create custom and commodity applications for the scenario
- Competitors are judged on their report, a "presentation to management," and their interactions with "the client" during the test.

Real World Examples

Scenarios in the past have included hospitals, elections, corporate networks:

- Average between 30 and 50 hosts per team
- Average between 5 and 20 custom applications in various languages

VDI
10.x.0.0/24

- Windows VDI — Windows 2016
- Kali VDI — Kali Linux
- Team DNS — Ubuntu 16

CORP
10.0.1.0/24

- DC01 — Windows 2016, MS ADDS, .100
- INFO01 — CentOS 7, Confluence, .101
- BILLING01 — Windows 2012, SimpleInvoices, .102
- CHAT01 — Ubuntu 16, Mattermost, .103
- WAREHOUSE01 — Windows 2012, MS SQL, .107
- TALENT01 — Ubuntu 16, Recruity, .104
- EXCH01 — Windows 2016, Exchange / OWA, .105
- ITUTIL01 — Ubuntu 16, Freeciv / Custom Apps, .106
- WS01 — Windows 2012, Desktop, .201
- WS02 — Windows 2012, Desktop, .202
- WS03 — Windows 2012, Desktop, .203
- WS04 — Windows 2012, Desktop / VNC, .204

DEV
10.0.2.0/24

- VPN01 — FreeBSD 11, OpenVPN, .250
- DC02 — Windows 2016, MS ADDS, .100
- CODE01 — Ubuntu 16, Gitlab, .50
- BUILD01 — Ubuntu 16, Gitlab-CI, .51
- DEBUG01 — CentOS 7, ELK Stack, .101
- BACKUPS01 — Windows 2012, CIFS / FTP, .102
- DEV01 — Ubuntu 16, Desktop, .1
- DEV02 — Ubuntu 16, Desktop, .2
- DEV03 — Ubuntu 16, Desktop, .3

PROD
10.0.100.0/24

- LOAD01 — AWS Linux, nginx Proxy, .200
- LOAD02 — AWS Linux, nginx Proxy, .201
- LOAD03 — AWS Linux, nginx Proxy, .202
- DC03 — Windows 2016, MS ADDS, .100
- ANALYTICS01 — Ubuntu 16, Piwik PHP, .102
- JIRA01 — Ubuntu 16, JIRA, .101
- WWW01 — CoreOS, nginx, .210
- PORTAL01 — Ubuntu 16, Portal Web App, .211
- DB01 — Ubuntu 16, PostgreSQL, .150
- DB02 — Ubuntu 16, MySQL, .151

GAME ADMIN
10.0.0.0/24

- LaForge — Ubuntu 16
- Splunk
- ELK Monitor — Ubuntu 16

SEC
192.168.44.0/24

- VPN02 — FreeBSD 11, OpenVPN / SSH, .250
- DC04 — Windows 2016, MS ADDS, .100
- MGMT01 — Ubuntu 16, SALT, .101
- MON01 — Ubuntu 16, GreyLog, .102
- AGG01 — Ubuntu 16, LogStash, .103
- VAULT01 — Ubuntu 16, Knox App, .104
- CA01 — Ubuntu 16, CFSSL, .105

ENET
192.168.254.0/24

- EGW01 — Ubuntu 16, SSH / Nginx RProxy, .250
- REGISTRY01 — Ubuntu 16, Registration App, .50
- BOOTH01 — Windows 2012, Voting App (Team), .20
- BOOTH02 — Windows 10, Voting App (Live), .21
- BOOTH03 — Windows 2012, Voting App (Script), .22
- VAPI01 — Ubuntu 16, Vote API App, .100
- VDB01 — Ubuntu 16, MySQL, .101
- VADMIN01 — Ubuntu 16, Election Monitor, .102
- KEYS01 — Ubuntu 16, Key Server, .103

PAIN IN MY INFRA

Needs Infrastructure as Code

The infrastructure needs to be accessible to multiple developers of varying skill sets. Must be debuggable, auditable, and developed across the team.

Needs to Scale Based on Teams

Each team needs an identical copy of the company network.

Think 200k+ lines of Terraform code per team each year.

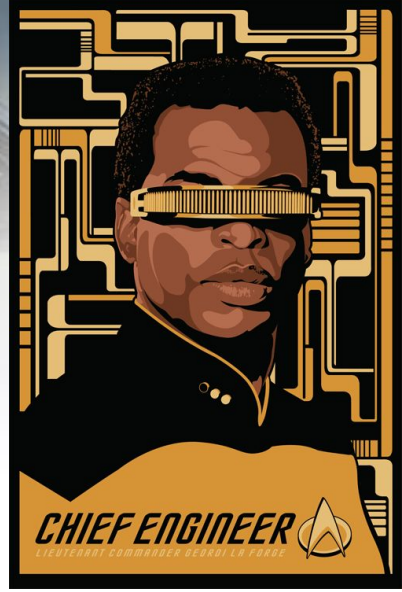
Needs to have a low learning curve for volunteers

Traditional DevOps tools have a steep learning curve. We need volunteers to contribute in their area of expertise, without learning a new language.

We need to develop a completely new environment every year.

THE SOLUTION?

!LAFORGE!



Competition Scripting

Each host and network is a configuration file and collection of scripts.

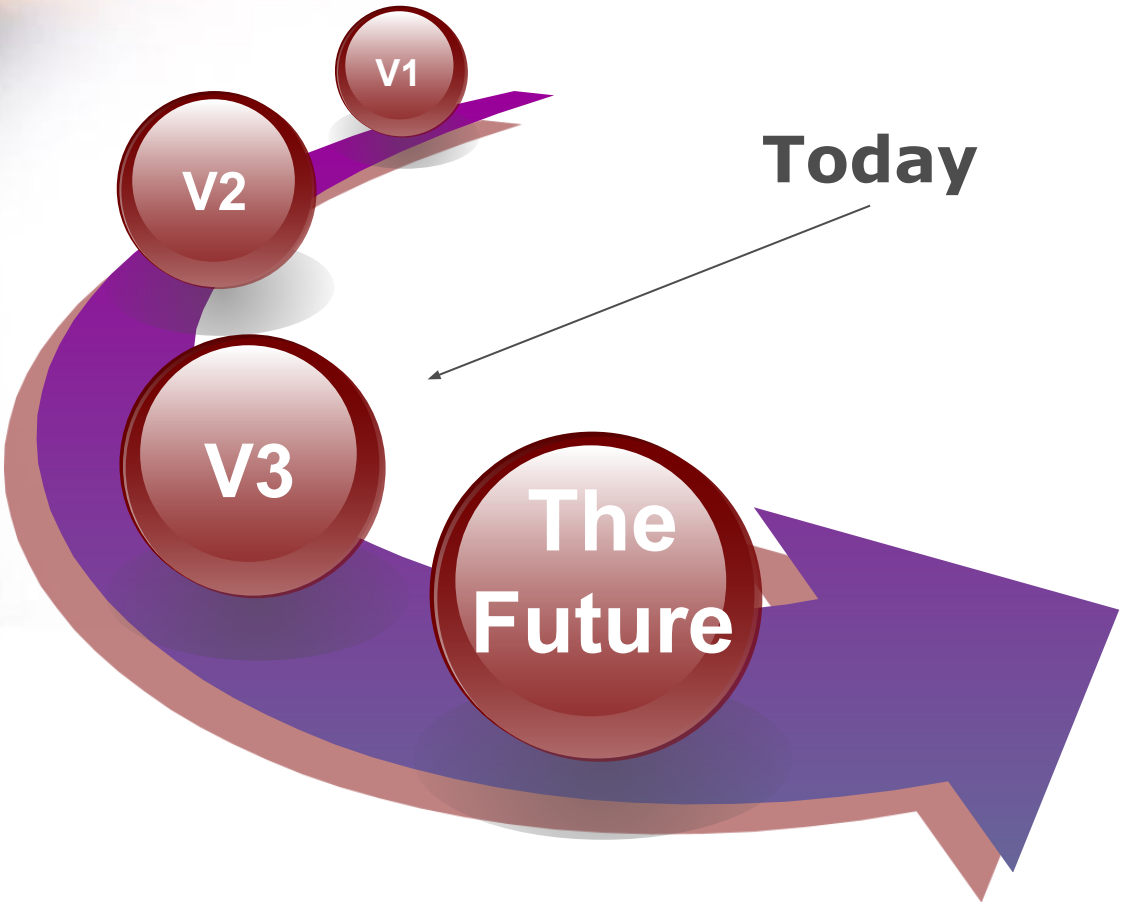
Developer Coordination

Developers can split tasks on hosts, or by script. They get unique environments to test in so they don't stomp all over each other.

Corporate and Competitor Systems

The entire competition, from servers to applications to competitor jump boxes is contained within Laforge.

PAIN IN MY INFRA



- v1 - **Ruby** script that leveraged **ERB templating** to write out **Terraform code**. Hard to port.
- v2 - Re-written in **Golang** and uses **YAML** files configuration. Produces a **monolithic Terraform script** for the entire competition.
- v3 - **Re-written again.** (Still Golang). YAML and monolith are gone. Custom config language, modularized for easily adding new features.

FEATURES

Build Once, Replicate Forever

You only build a single competition set, and then LaForge creates an copy per team

Uses scripts, not images, allowing you to reuse year to year and simplify dev.

Universal Config Language

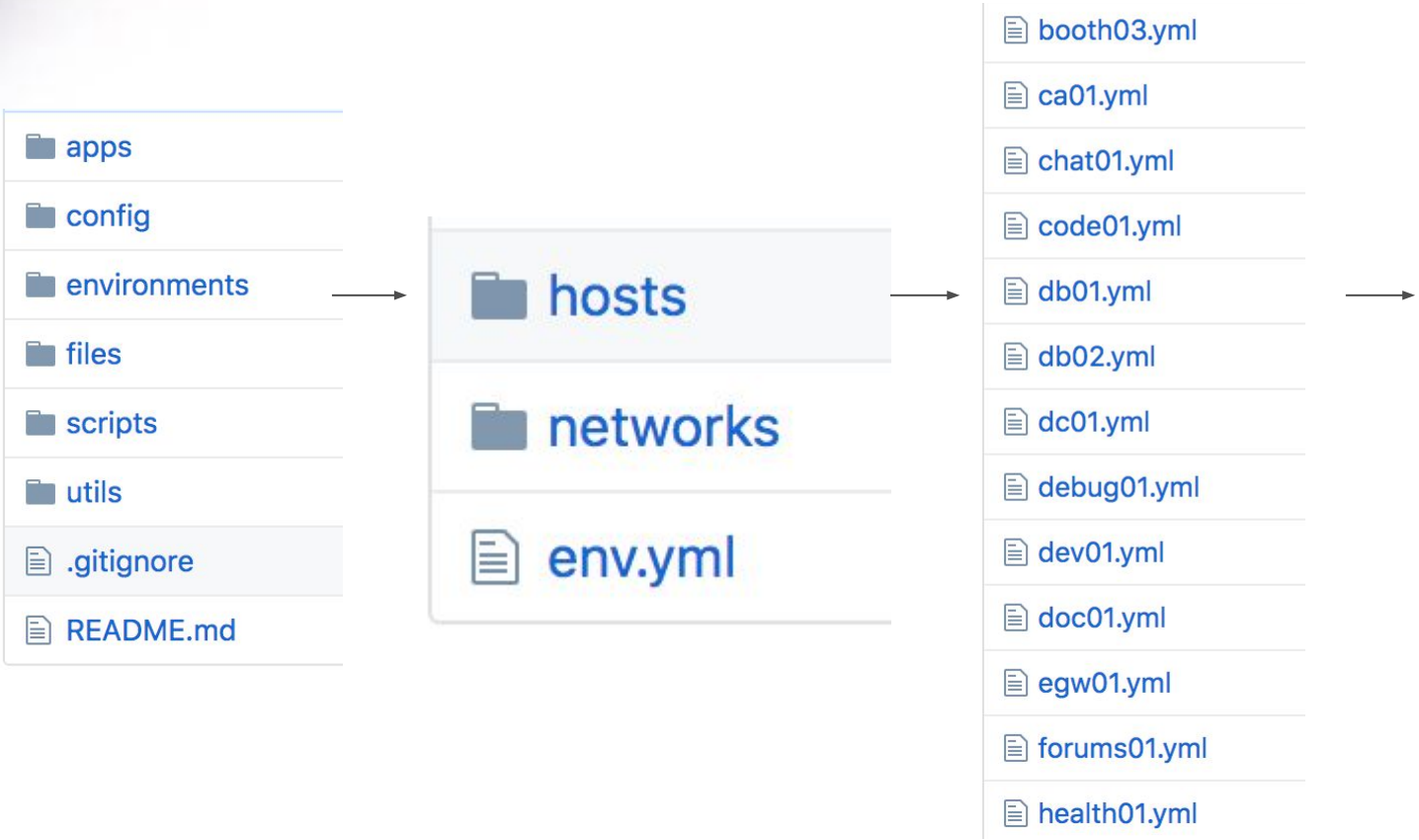
New Laforge has been designed to use a markup similar to UCL, that provides far more flexibility than YAML. Goodbye Whitespace problems!.

Native Scripting

Build hosts in the languages you already know (bash, batch, powershell, etc.)

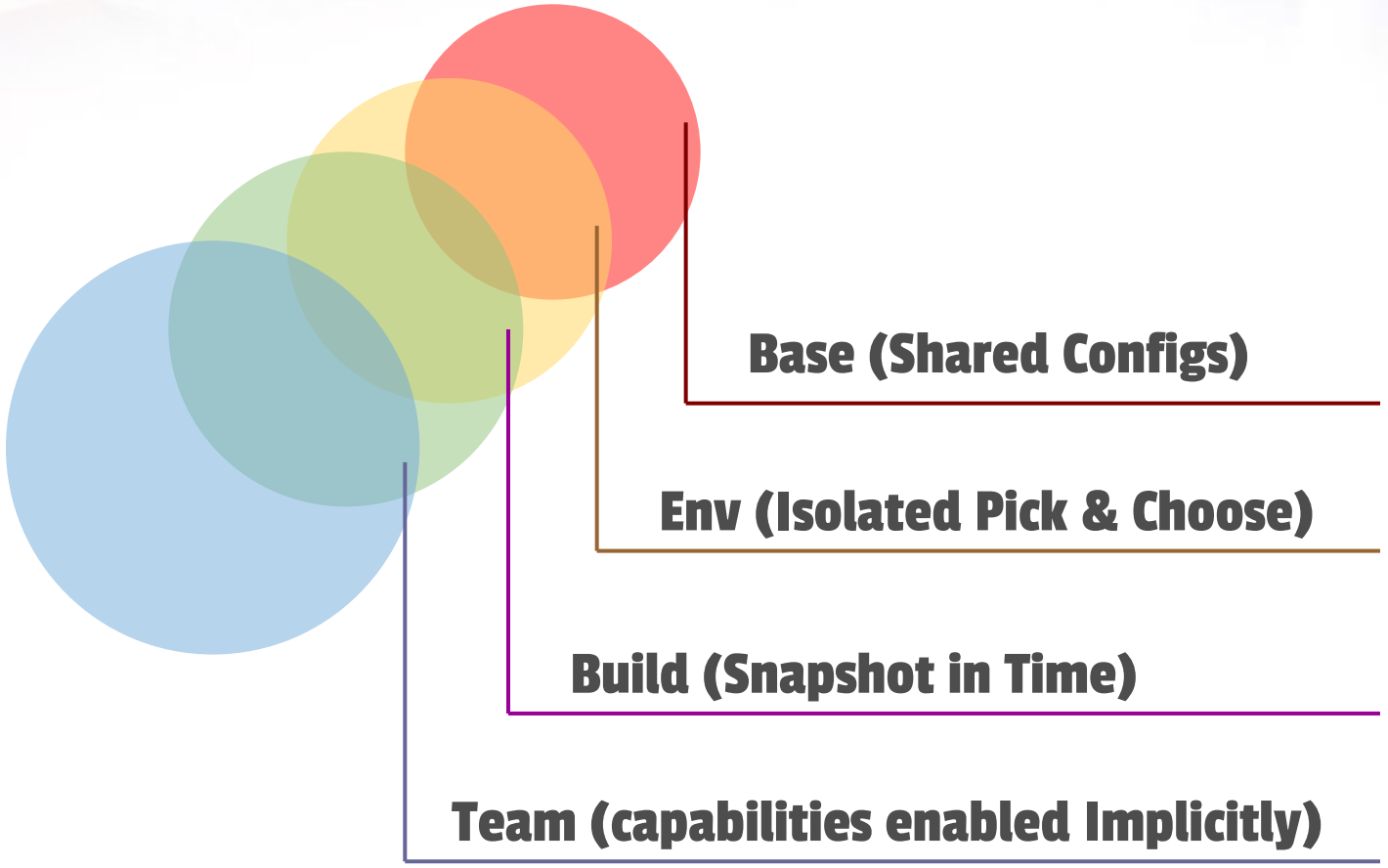
Plug and play scripts on hosts and re-use all the things!

CONFIGURATION AS CODE




```
hostname: health01
os: ubuntu
ami:
last_octet: 245
internal_cnames: []
external_cnames:
  - election-status
instance_size: c4.xlarge
scripts:
  - health01.sh
  - salty_logs.sh
  - ubuntu_motd.sh
  - suricata_linux.sh
  - splunk_basic.sh
  - osquery_deb.sh
  - domain-join-debian.sh
user_groups: []
variables:
  example: variable
public_tcp:
  - 80
  - 1001
  - 443
  - 514
  - 587
  - 25
public_udp:
  - 514
files:
  happy.txt: /tmp/happy.txt
dependencies:
  - host: dc01
    network: corp
```

MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE



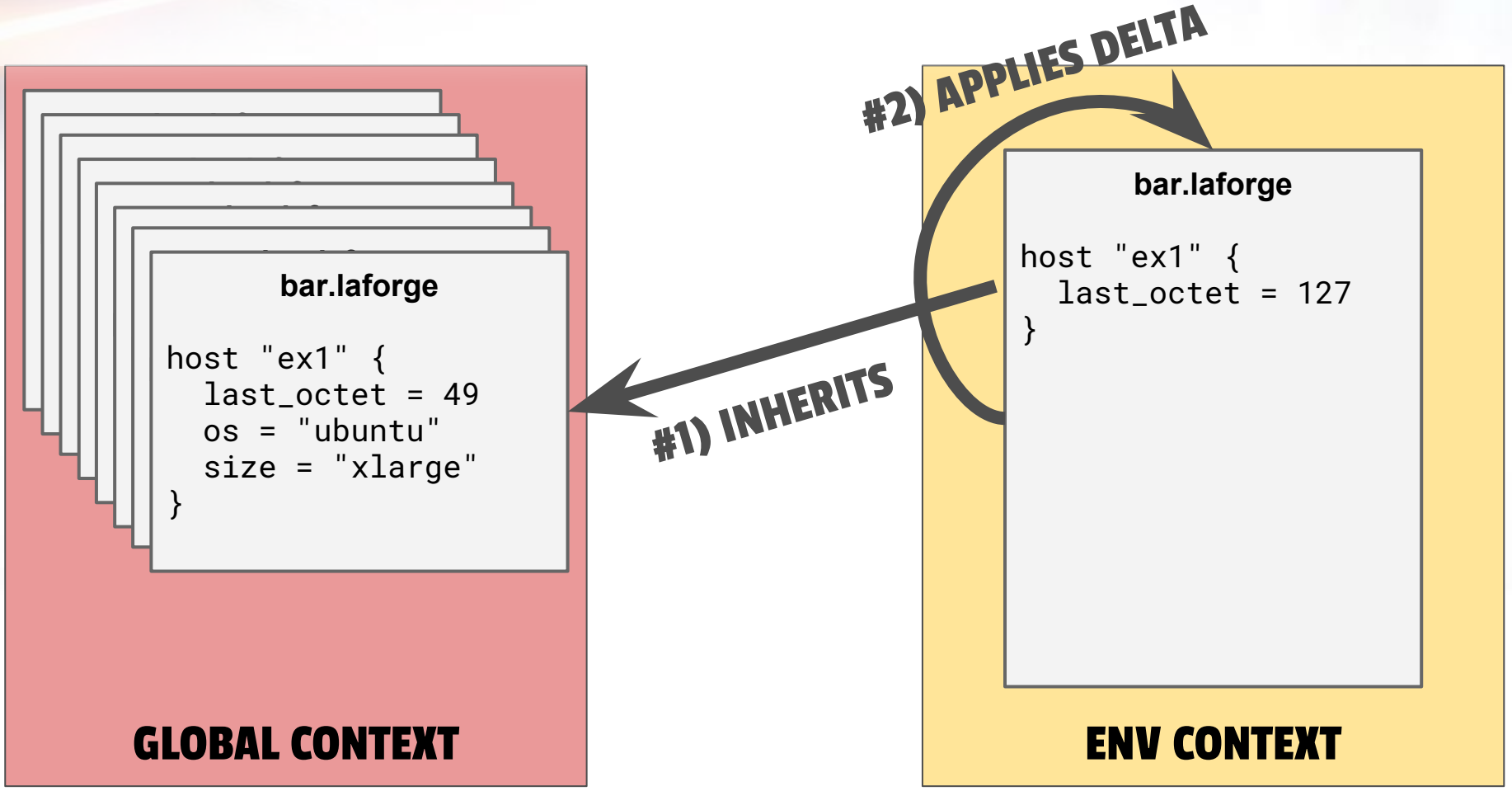
MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE

MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE

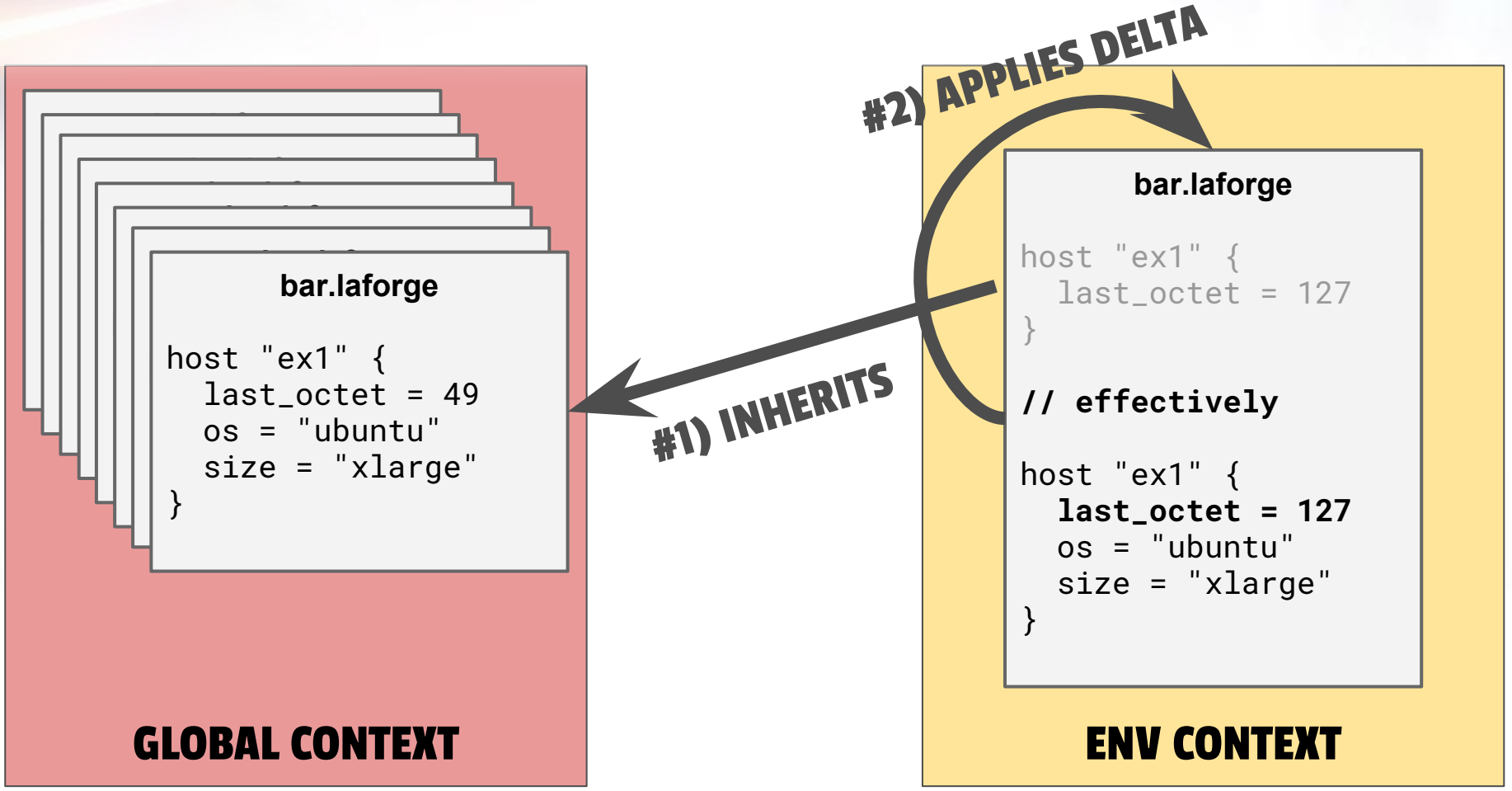
```
09:06:29 urca/flint @rekt <2.5.0@base> $ laforge status
[LAFORGE:cli]  INFO Current Context Level
 (0) TeamContext
 (1) BuildContext
 (2) *CURRENT* EnvContext
 (3) BaseContext
 (4) GlobalContext
09:06:33 urca/flint @rekt <2.5.0@base> $
```

Context decides implicitly what files get precedence in the dependency graph.

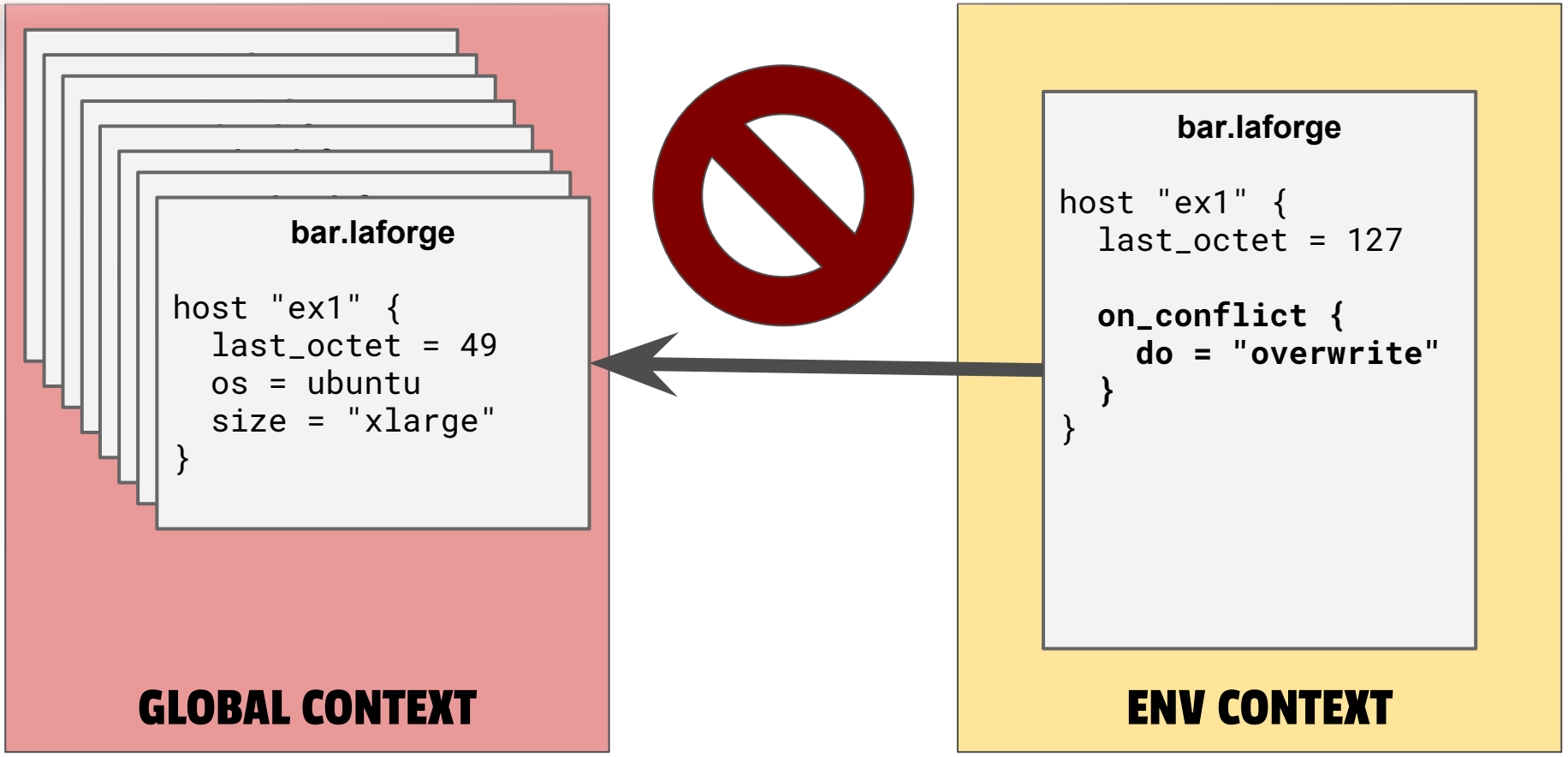
MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE



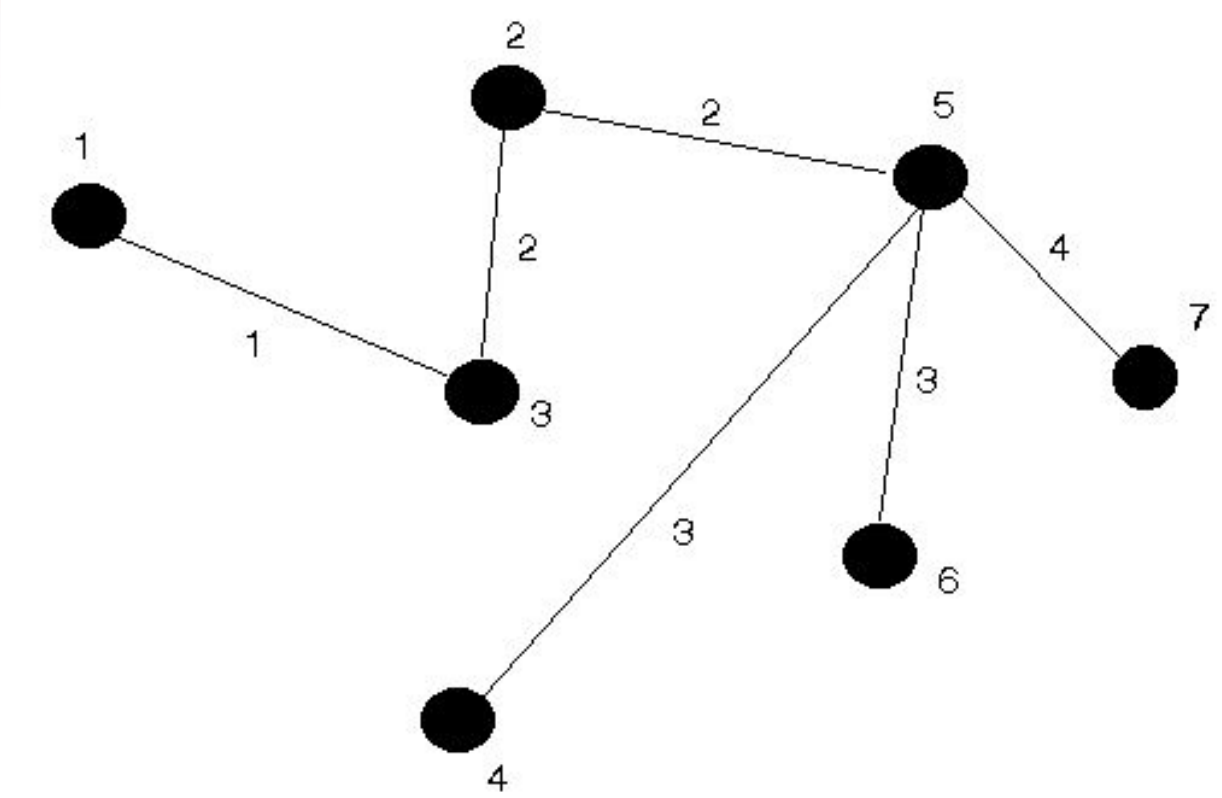
MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE



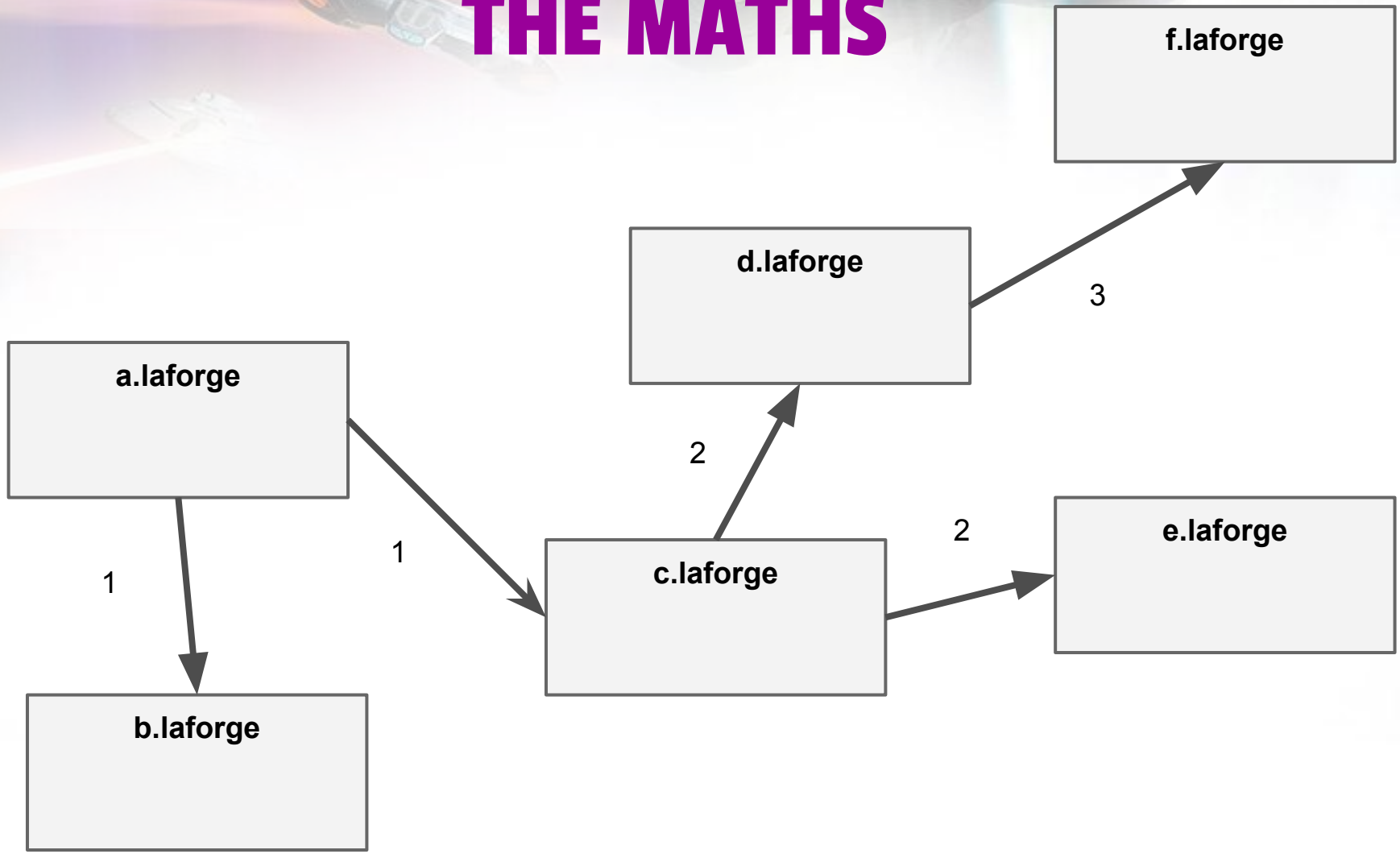
MULTI-DIMENSIONAL, NON-DESTRUCTIVE INFRASTRUCTURE



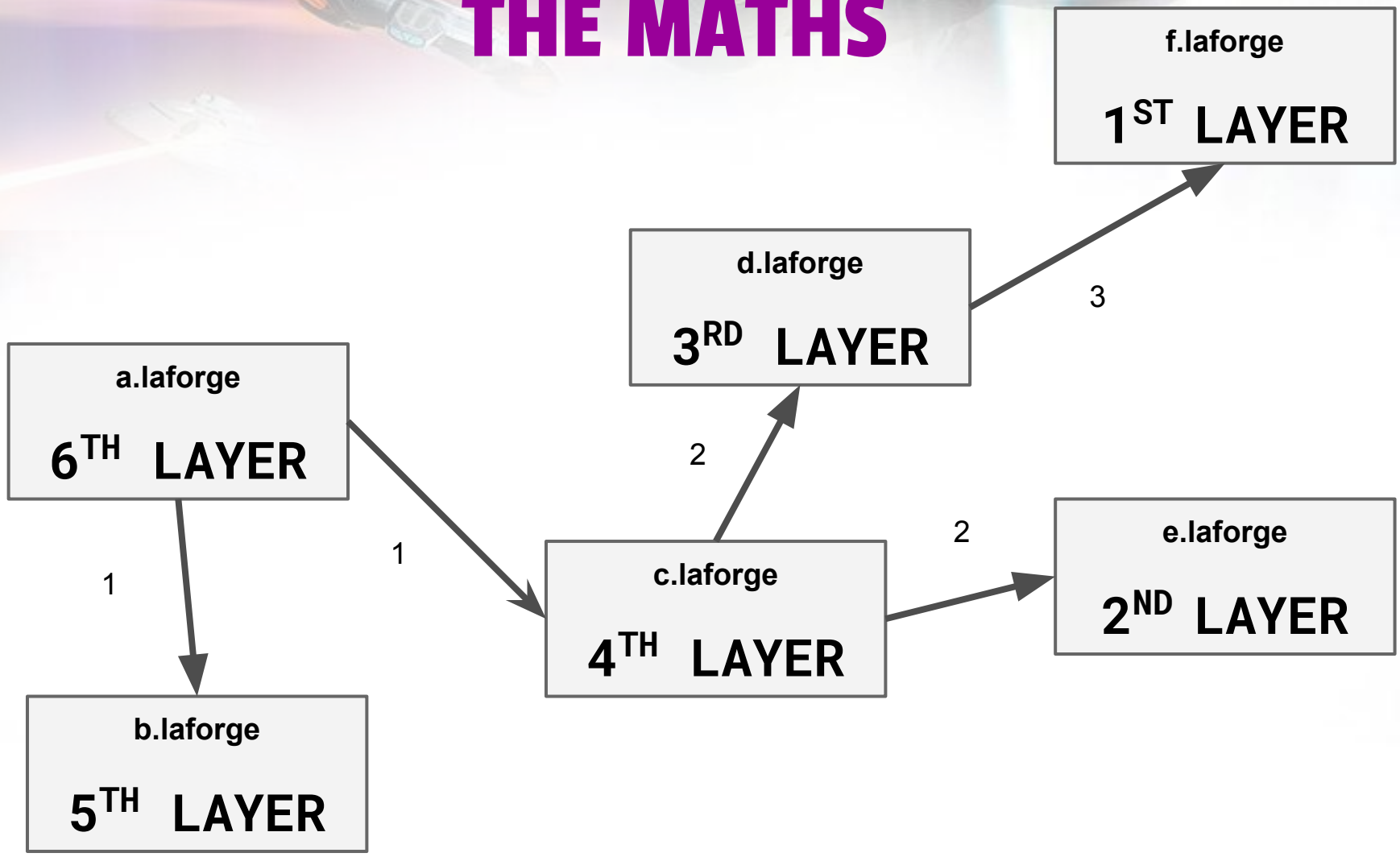
THE MATHS



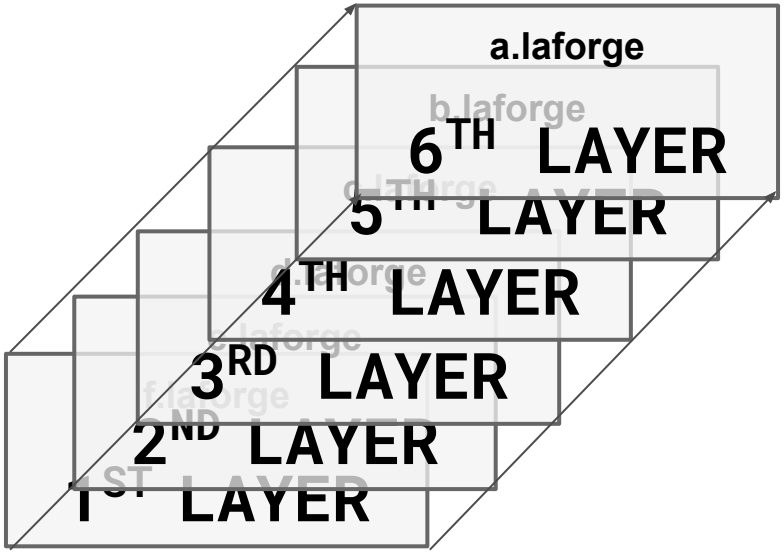
THE MATHS


f.laforge
d.laforge
3
a.laforge
2
1
1
2
c.laforge
e.laforge
b.laforge

THE MATHS


f.laforge
1ST LAYER
d.laforge
3RD LAYER
3
a.laforge
6TH LAYER
2
1
1
2
c.laforge
4TH LAYER
e.laforge
2ND LAYER
b.laforge
5TH LAYER

THE MATHS


a.laforge
6TH LAYER
5TH LAYER
4TH LAYER
3RD LAYER
2ND LAYER
1ST LAYER

ALPHA PREVIEW VERSION LIVE!
(we're confident lol)

Multi-Dimensional

Treats your environment very much like a git repo, removes flat files, adds includes, and uses more than just Terraform.

LaForge Config Syntax

Straightforward and flexible, implements imports, named scopes, and is used to configure everything about your environment.

Pull requests are welcome!

If you like this project and would like to help us in development:

https://github.com/gen0cide/laforge

TL;DR

LaForge is a modern DevOps tool specifically for creating real-world scenarios and competition environments.

Allows the Rapid Development of Competition Networks

Build reusable and scalable competition environments.

Coordinate Development

LaForge is extensible, allowing multiple developers, multiple hosts, and multiple outputs.

The tool is already out there!

Stop listening to us! Download LaForge and hack with it!


QUESTIONS?
END
github.com/genOcide/laforge
nationalcptc.org
BSIDESLV 2018