Xen Project Contributor Training
Part 4 : Culture

Lars Kurth
Community Manager, Xen Project
Chairman, Xen Project Advisory Board
Director, Open Source Business Office, Citrix

lars_kurth

Content

Theory: Open Source Flywheel

The demands on what vendors and users want from Xen Project is changing using the Flywheel to illustrate

The project has a recent history of change

Example: The history of the Security Vulnerability Management Process

Other examples of recent and ongoing changes

New demands on the project: New Features/Community Growth vs. Review Process and Review Capacity

New demands on the project: New Features/Community Growth vs. Quality and Security

Feature Lifecycle Management and Documentation

Theory:

Open Source Flywheel

Users

Feedback, Engagement
Trust, Passion, Media Coverage

Open Source Development Model

Tools, Process, Culture
Option Value[1], Modularity

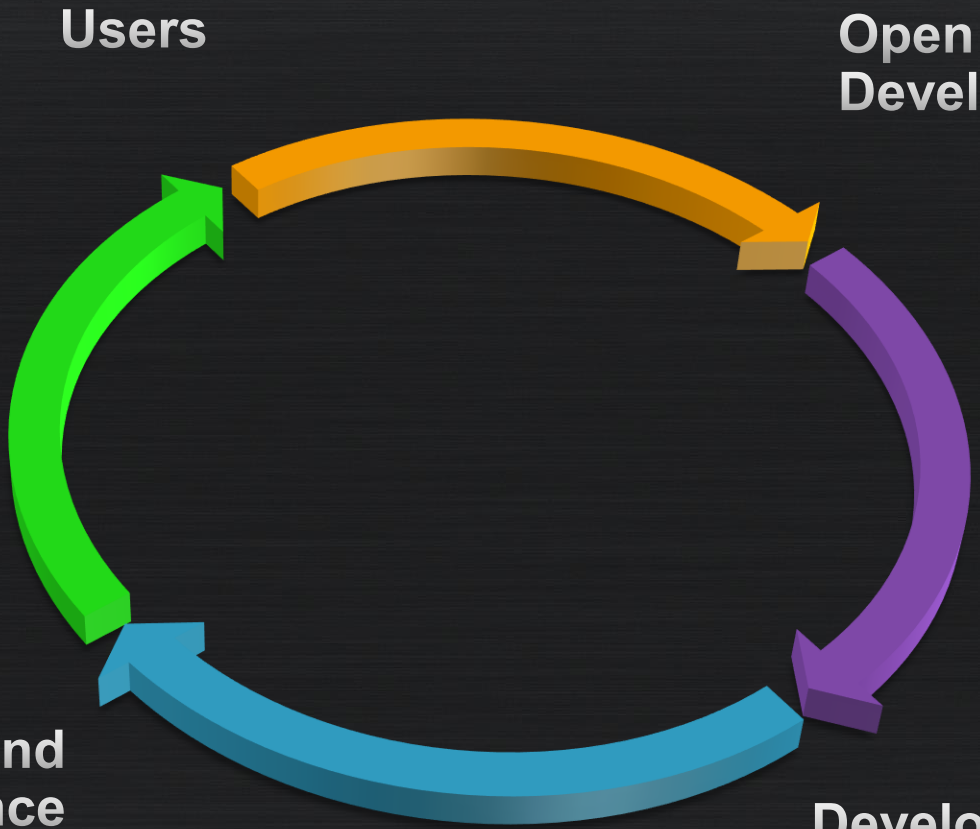
[1] bit.do/optionvalue

Product and Experience

Features, Quality
3rd Party Integrations

Development Activity

Contributions, Reviews,
Problem Solving, Leadership


Users
Open Source Development Model
Development Activity
Product and Experience

More
Users

More business opportunities and momentum

Open Source
Development Model

More
Development Activity

Lower development cost

Better
Product and
Experience

Lower deployment cost and risk


More
Users
More business opportunities and momentum
Open Source Development Model
Community Growth
Better
Product and Experience
Lower deployment cost and risk
More
Development Activity
Lower development cost

More Users

More business opportunities and momentum

Better Open Source Development Model

More efficiency and innovation

More Development Activity

Lower development cost

Better Product and Experience

Lower deployment cost and risk

War Stories:

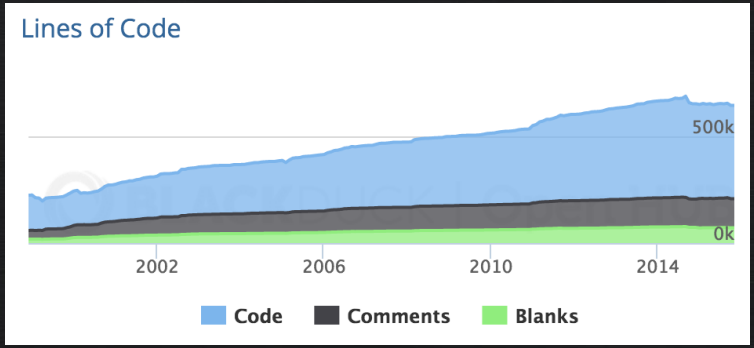
Tragedy of the Commons
(sort of)

Moyan Brenn @ Flickr

snoopsmouse @ Flickr

So what happened and why?

OpenSSL Stats



Source: Ohloh.net

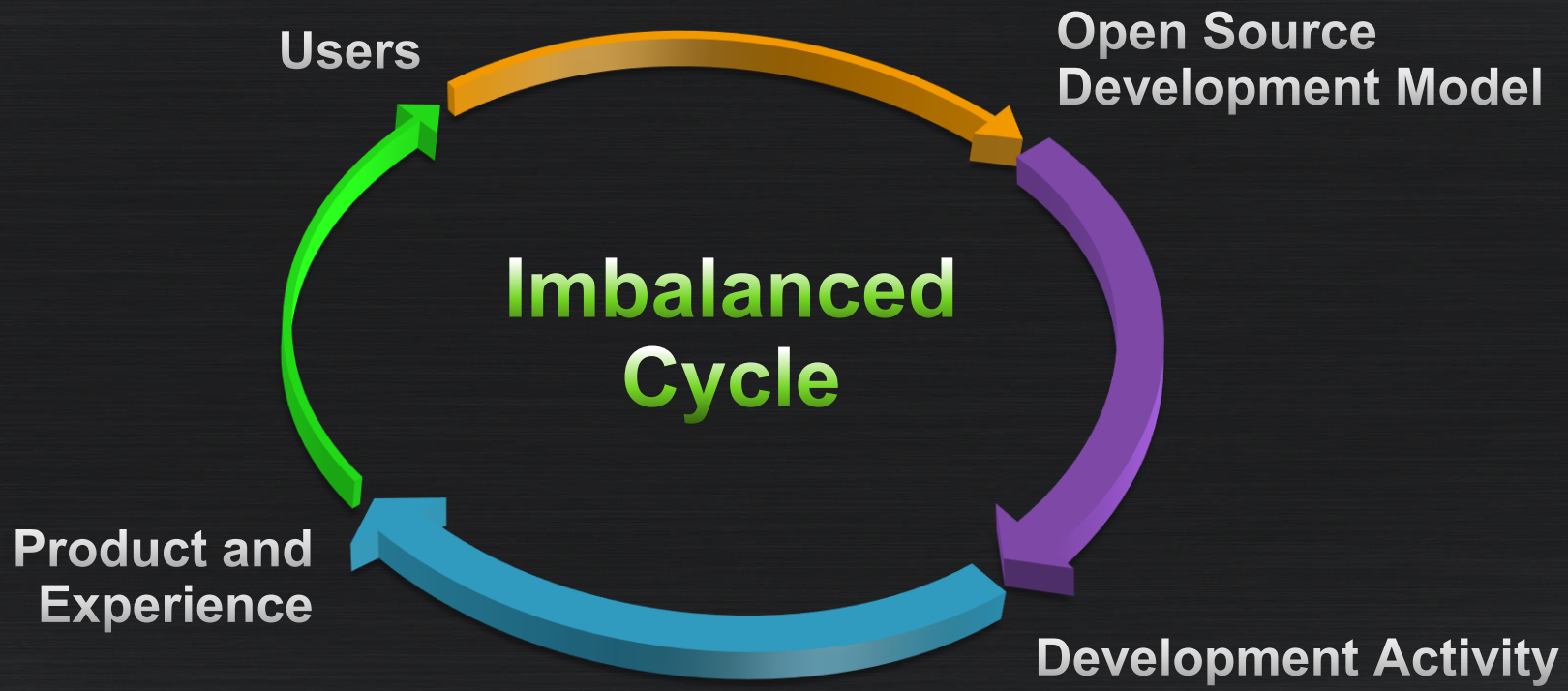
Prior to Heartbleed

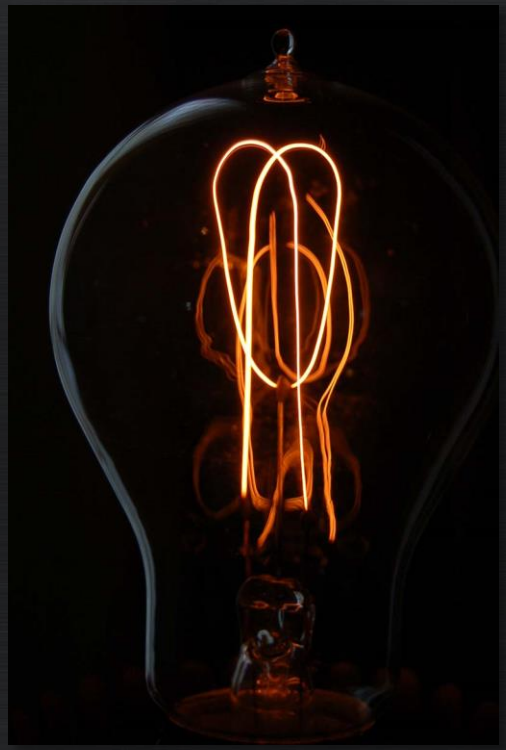
Growing Codebase

Static and small contributor base
1 person maintaining 100 KLoC = Underinvestment

Extremely large user base
Critical infrastructure component
Thus impact of Heartbleed is huge

Large user base did not translate into developer community growth

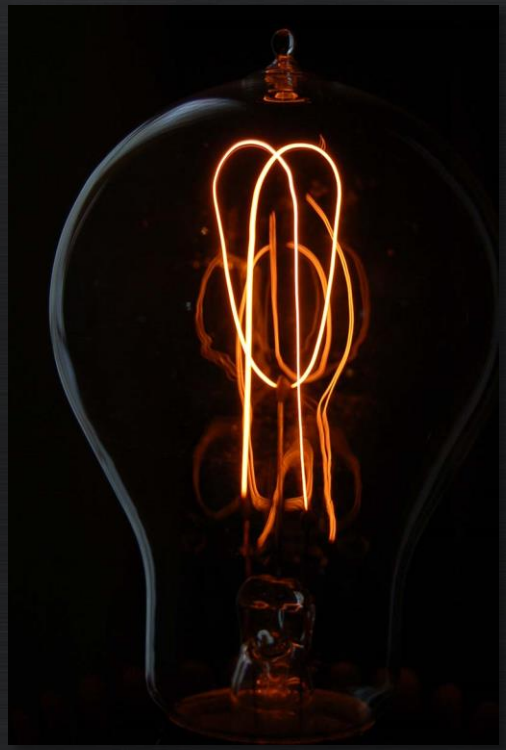

Users
Open Source Development Model
Imbalanced Cycle
Product and Experience
Development Activity



Lesson for Xen Project

Stay vigilant to sustain a balanced Flywheel

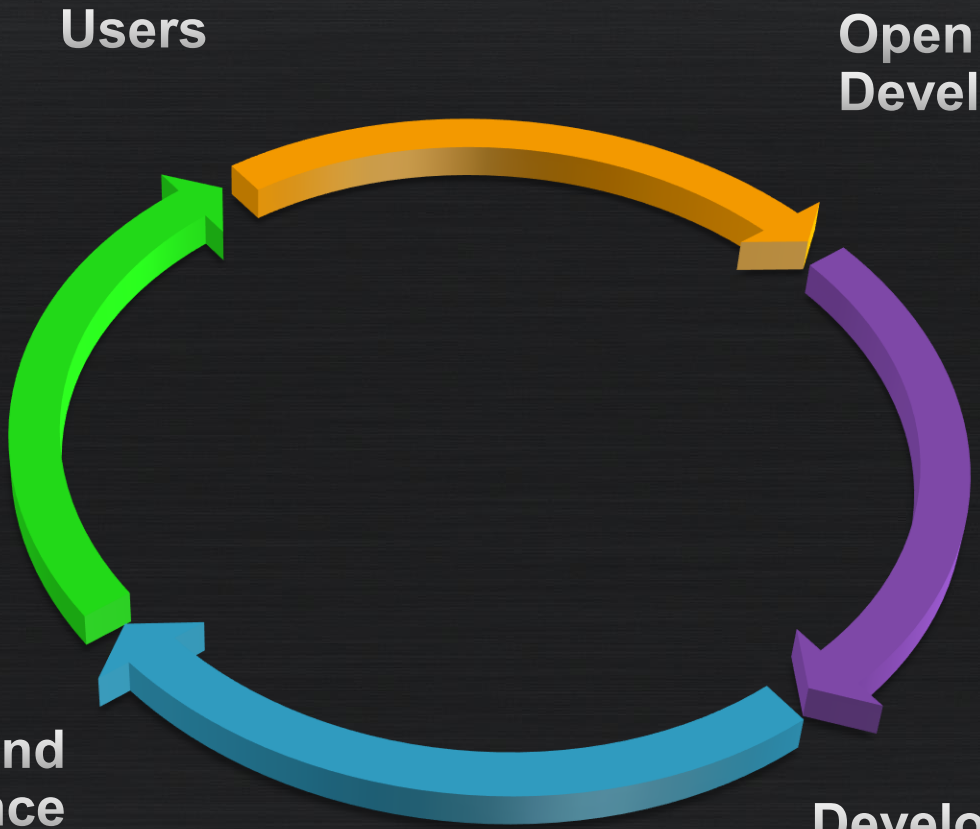
Vinovyn @ Flickr

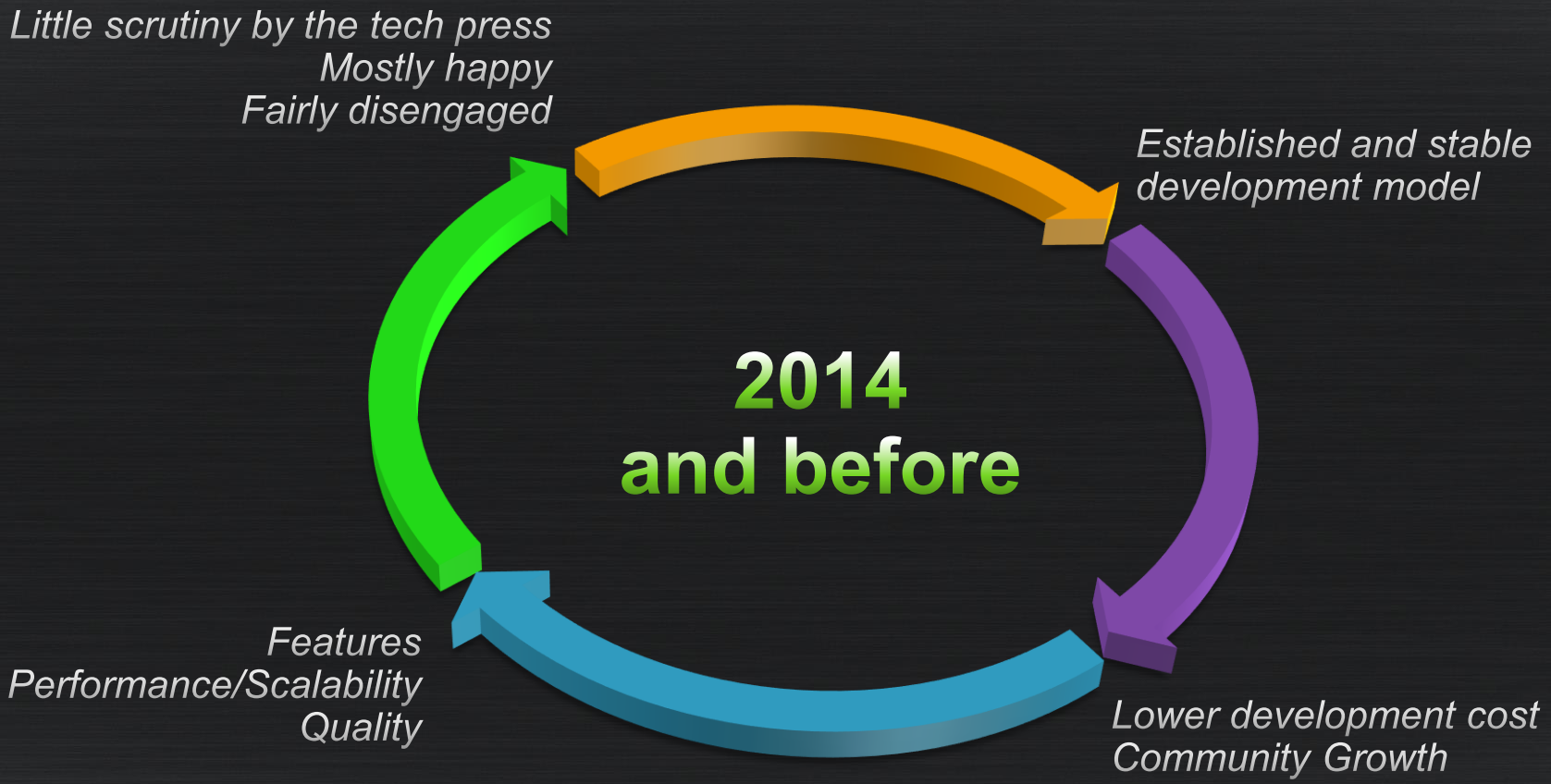


Drivers for Change

The Demands on what vendors and users want from Xen Project is changing

Vinovyn @ Flickr


Users
Open Source Development Model
Development Activity
Product and Experience


Little scrutiny by the tech press
Mostly happy
Fairly disengaged
Established and stable development model
2014 and before
Lower development cost
Community Growth
Features
Performance/Scalability
Quality

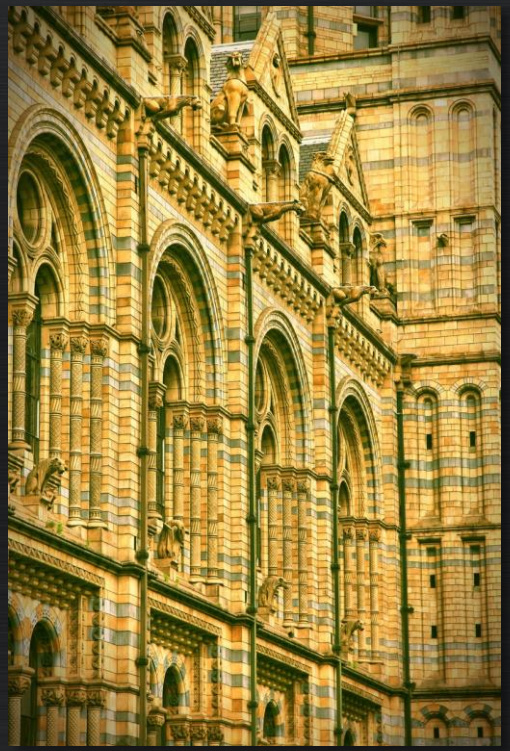
Huge amount of scrutiny *by the tech press (security, process, releases, …)*
Some users unhappy *(status quo vs. change)*
Vocal users and vendors *(the odd "rant")*

Community is forced to change:
Training, Test Lab(s), Review vs. Features, Security Management Process, Security vs. Features, Release Process, …

Features
Performance/Scalability
Higher *Quality*
Security
Usability / Integrations

More competition (e.g. Containers, Docker, …)

Lower development cost
Community Growth *(not at all cost)*
New Players: *Security, Embedded, …*
New Regions: *e.g. China & Ukraine*
More aggressive product roadmaps



Xen has a history of recent change

External factors are accelerating the amount of change

Example:

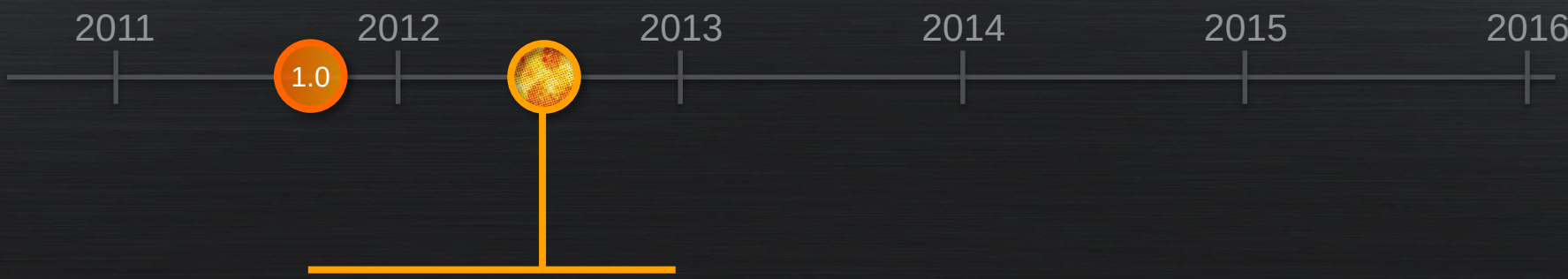
Evolution of
Xen Project Security Vulnerability Process

xenproject.org/security-policy.html

V1.0 : Modelled on Debian

Goals:
Allow fixing, packaging and testing;
Allow service providers to prepare (but not deploy) during embargo

Pre-disclosure:
Membership biased towards distros & large service providers
No predefined disclosure time

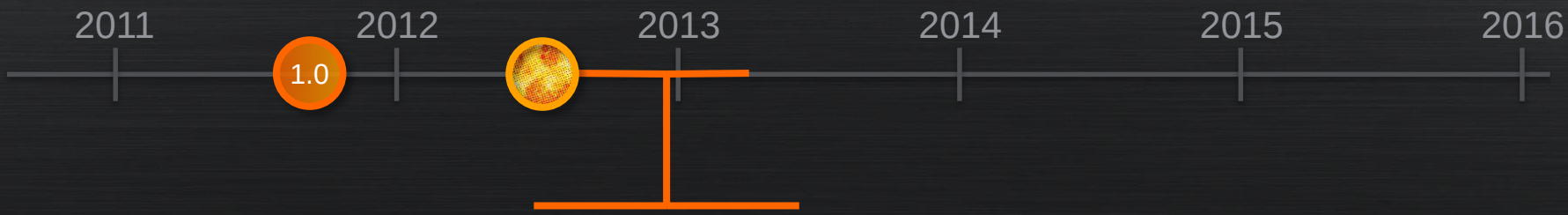


July 2012: CVE-2012-0217, Intel SYSRET

Affected FreeBSD, NetBSD, Solaris, Xen and Microsoft Windows

A large pre-disclosure list member put pressure on key members of the Xen Project Community to get an embargo extension

They eventually convinced the discoverer to request an extension



Community Consultation to improve our process

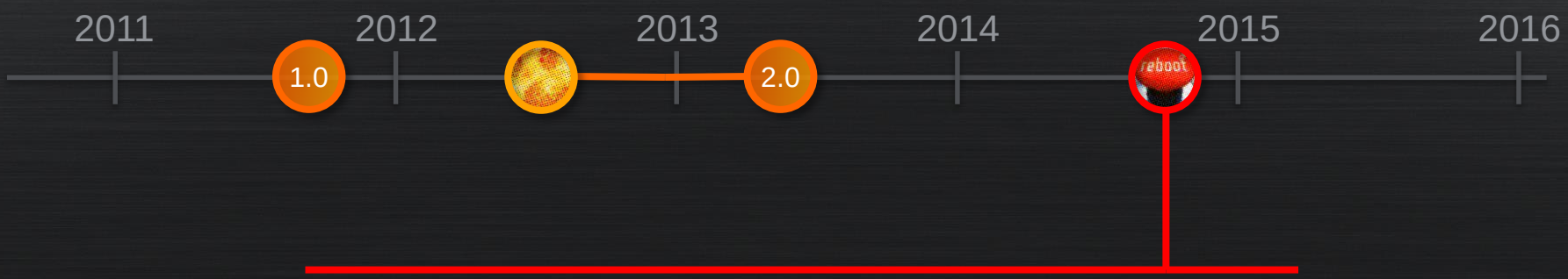
Centered on:

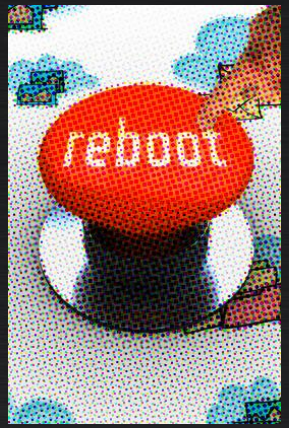
Predetermined disclosure schedule: 1 week to fix, 2 weeks embargo

Who should be allowed on the pre-disclosure list
Fairness issues between small and large service providers
Direct vs. indirect Xen consumers
The risk of larger pre-disclosure list membership

V2.0 : Clarifications

Strongly recommended disclosure schedule
Inclusive pre-disclosure list membership
Changes to application procedure (based on checkable criteria)





Sept 2014: CVE-2014-7118

Leading to the first Cloud Reboot

AWS pre-announced cloud reboot to their customers
Other vendors didn't.
Policy was interpreted differently by vendors.

This highlighted ambiguities in the project's security policy (what can/can't be said/done during an embargo)

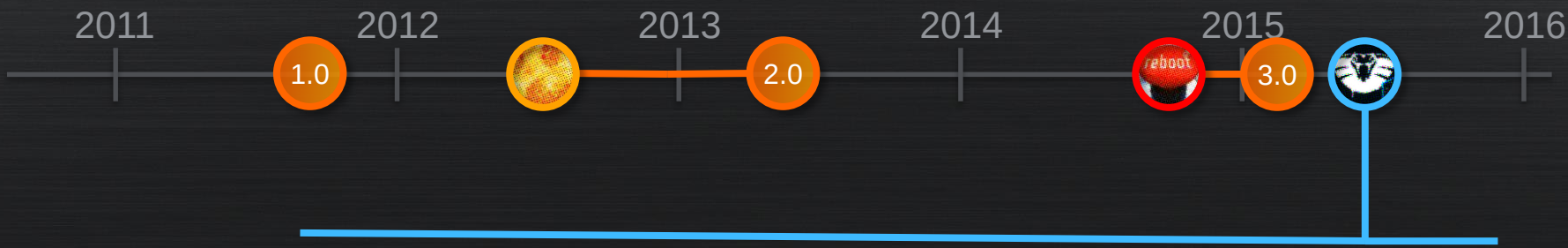
V3.0 : Deploy & Optimizations

Goals:

Allow fixing, packaging and testing
Allow service providers to prepare (and normally to deploy) during embargo

Pre-disclosure:

Clearer application criteria
Public application process (transparency)
Clear information on what is/is not allowed during an embargo (per XSA)
Means for pre-disclosure list members to collaborate



May 2015: CVE-2015-3456

First time we were affected by a branded bug

QEMU bug, which was handled by several security teams: QEMU, OSS Distro Security, Oracle Security & Xen Project

From a process perspective: were not able to provide a fix 2 weeks before the embargo date ended

Conducted XSA-133 Retrospective upon request

Process change: Earlier embargoed pre-disclosure without patches