Release Synchronization in Software Ecosystems (SECO)

Armstrong Foundjem, *Bram Adams*

MCIS — Laboratory
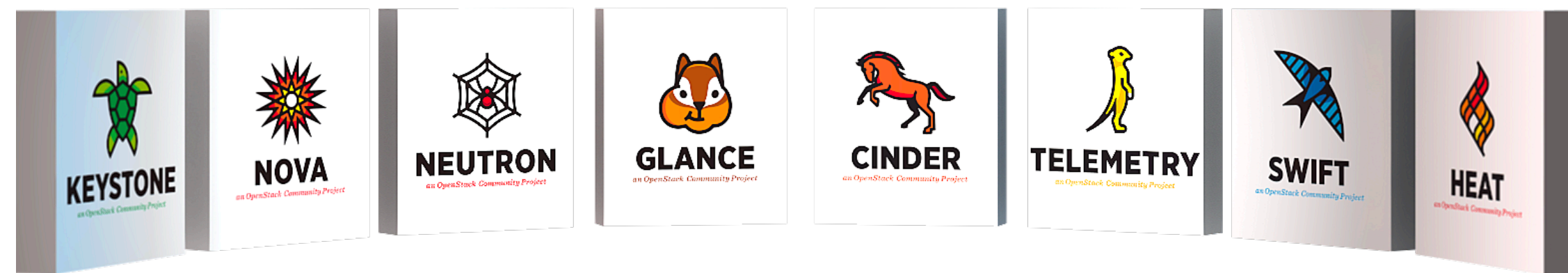
Queen's University, Kingston

Software Ecosystem (SECO): Set of Inter-dependent software projects with common goals that co-evolve in the same environment.

Example of SECOs

Lungu et al. "**The small project observatory: Visualizing software ecosystems.**" *Science of Computer Programming* 75.4 (2010)

Software Ecosystem (SECO): Set of Inter-dependent software projects with common goals that co-evolve in the same environment.

Example of SECOs

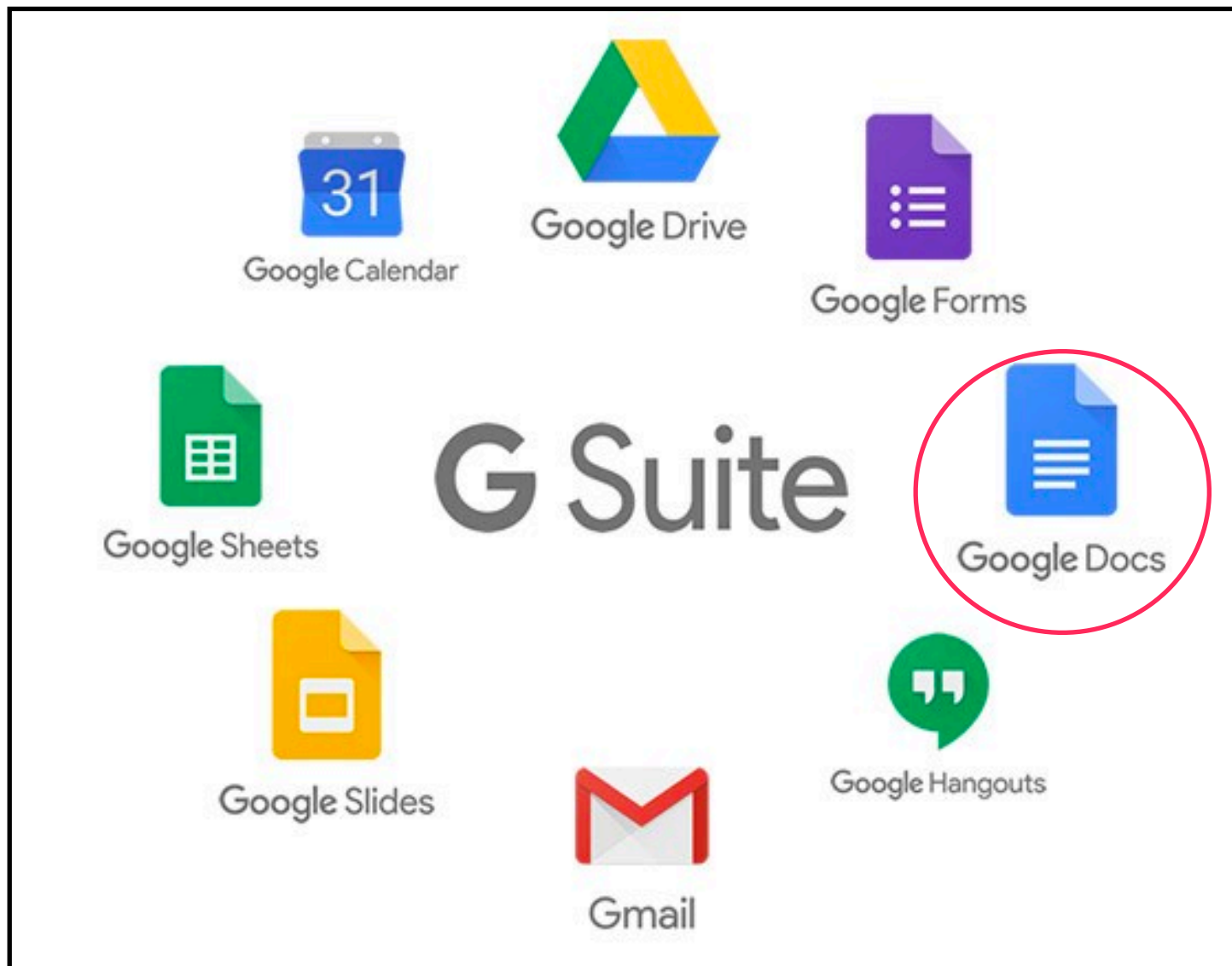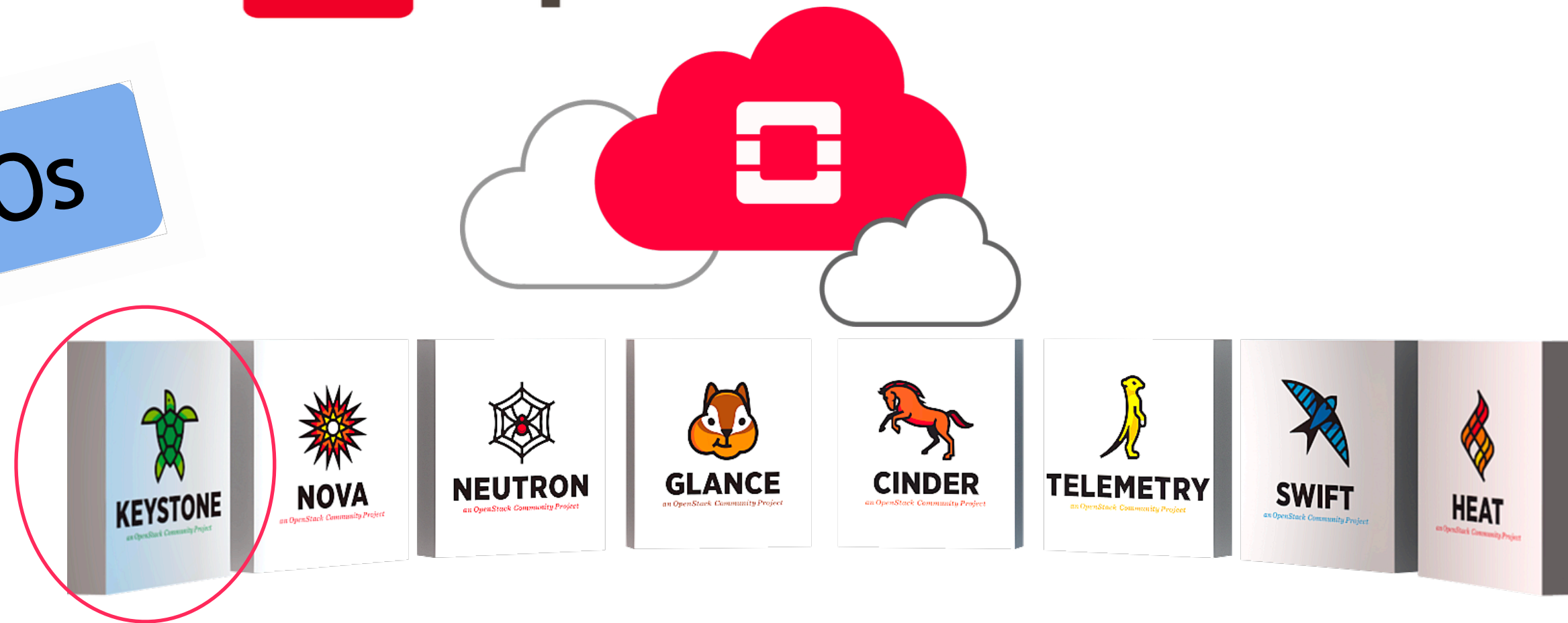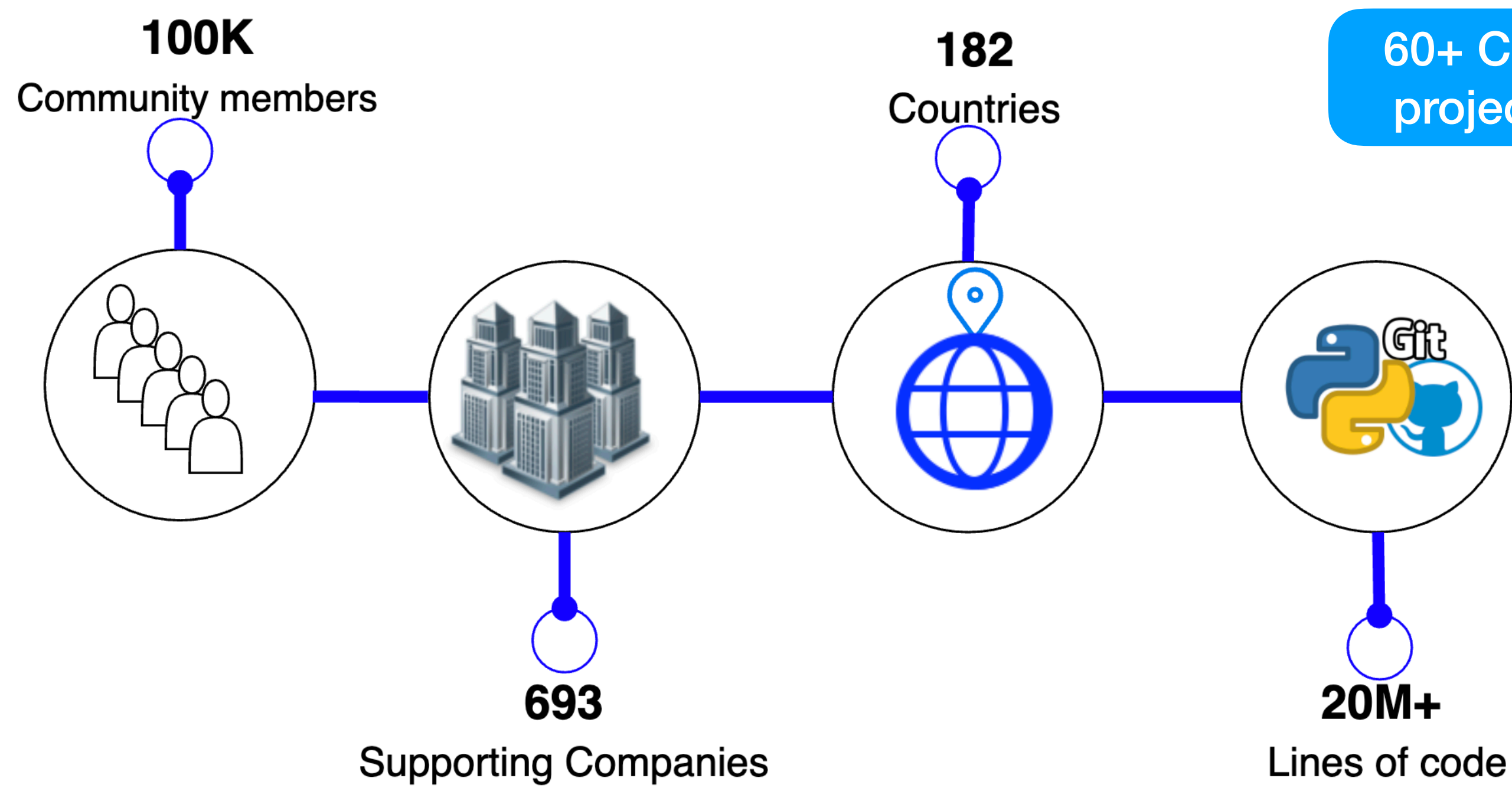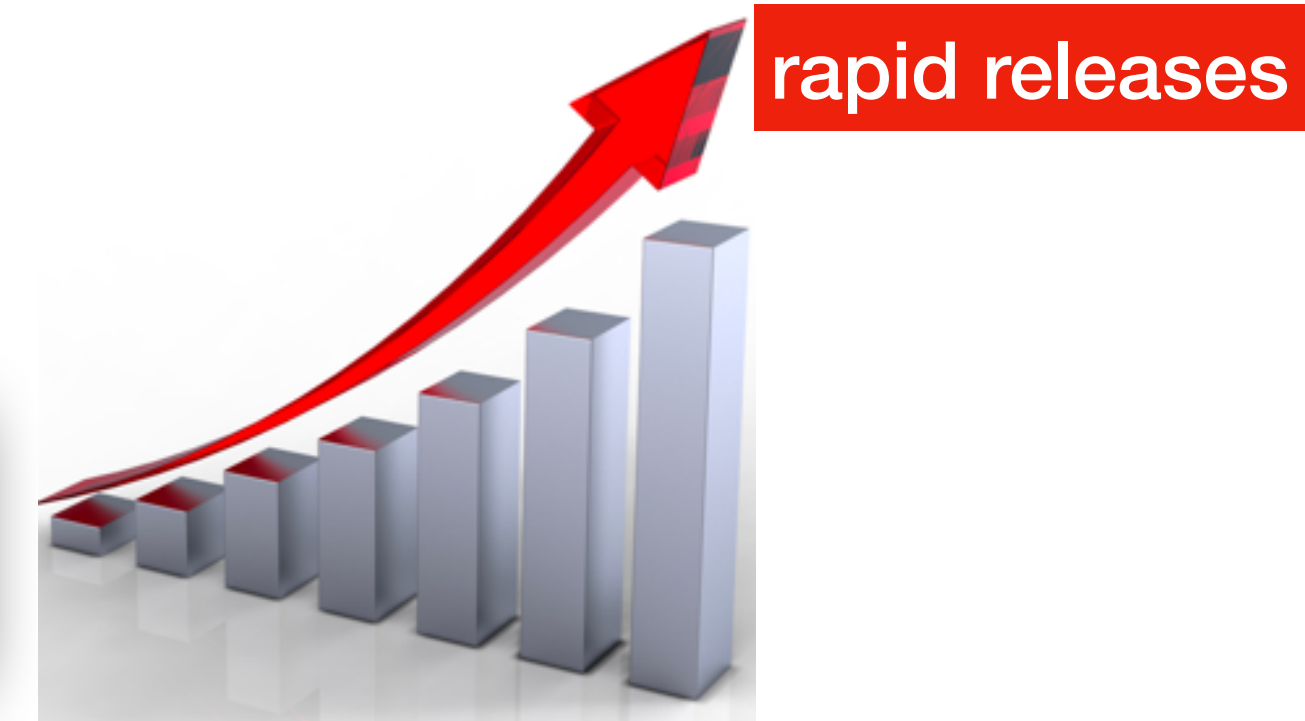Each project has separate road-map besides the coordinated SECO agenda

SECOs are Complex!

Why OpenStack?

rapid releases

100K
Community members

182
Countries

693
Supporting Companies

20M+
Lines of code

Centralized Release team with open access to data
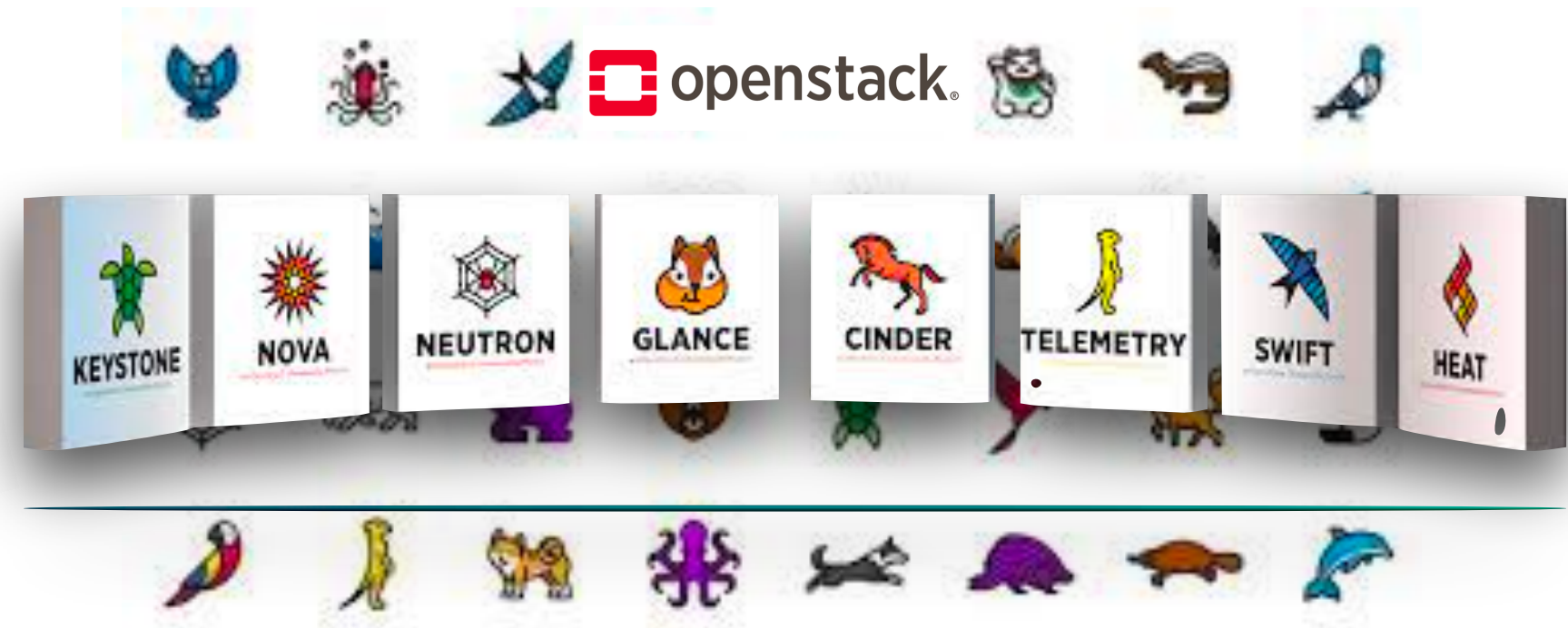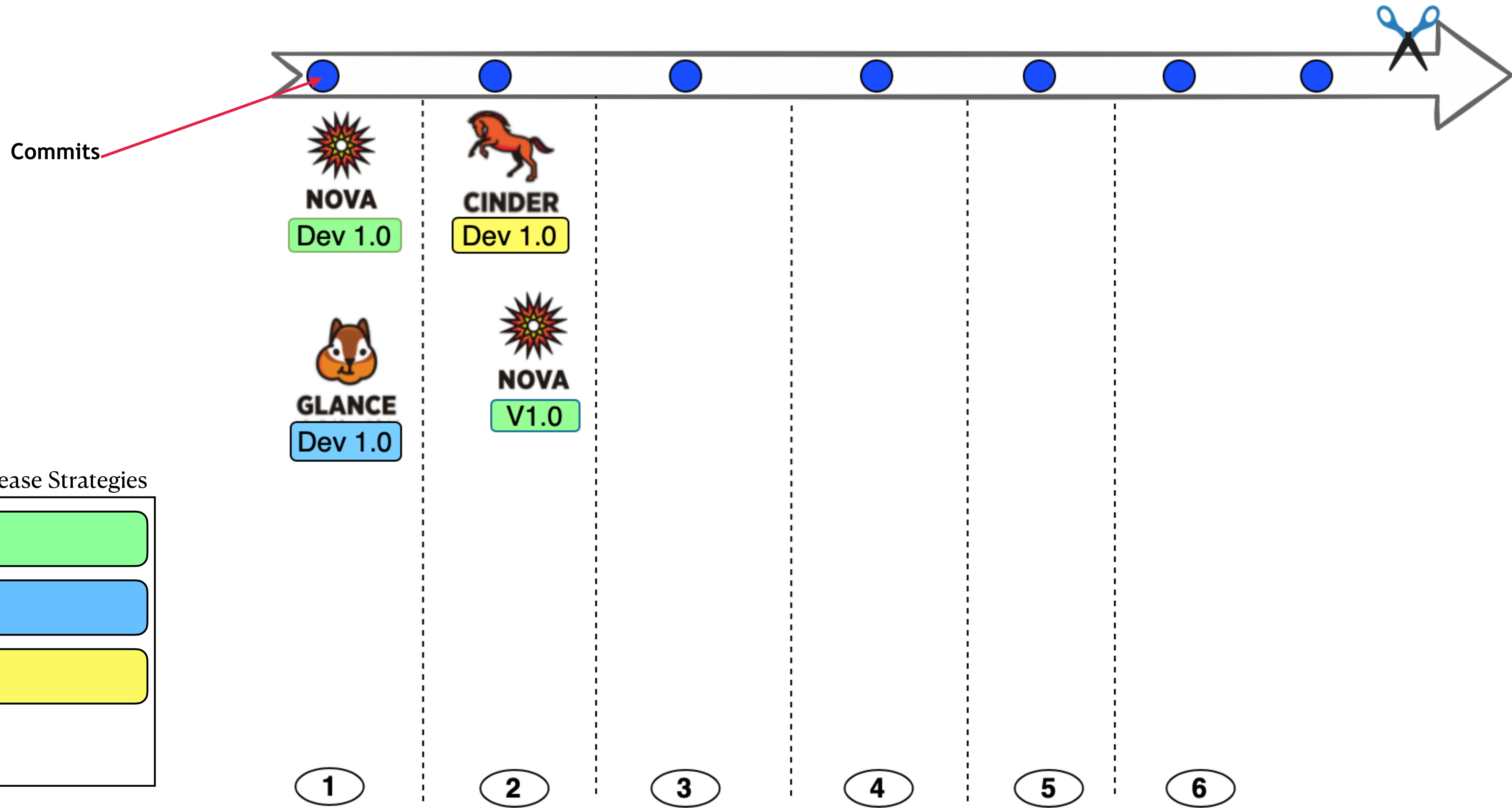
60+ Core projects

500+ Sub-projects

How to Release a SECO with each project having it's road-map?

Commits

NOVA
Dev 1.0

CINDER
Dev 1.0

GLANCE
Dev 1.0

NOVA
V1.0

Release Strategies

1   2   3   4   5   6

How to Release a SECO with each project having it's road-map?

Commits

Release Strategies

NOVA
Dev 1.0

CINDER
Dev 1.0

NOVA
Dev 1.1

GLANCE
Dev 1.1

GLANCE
Dev 1.0

NOVA
V1.0

KEYSTONE
Dev 1.0

KEYSTONE
V1.0

GLANCE
V 1.0

CINDER
V1.0

NOVA
V1.1

1 2 3 4 5 6
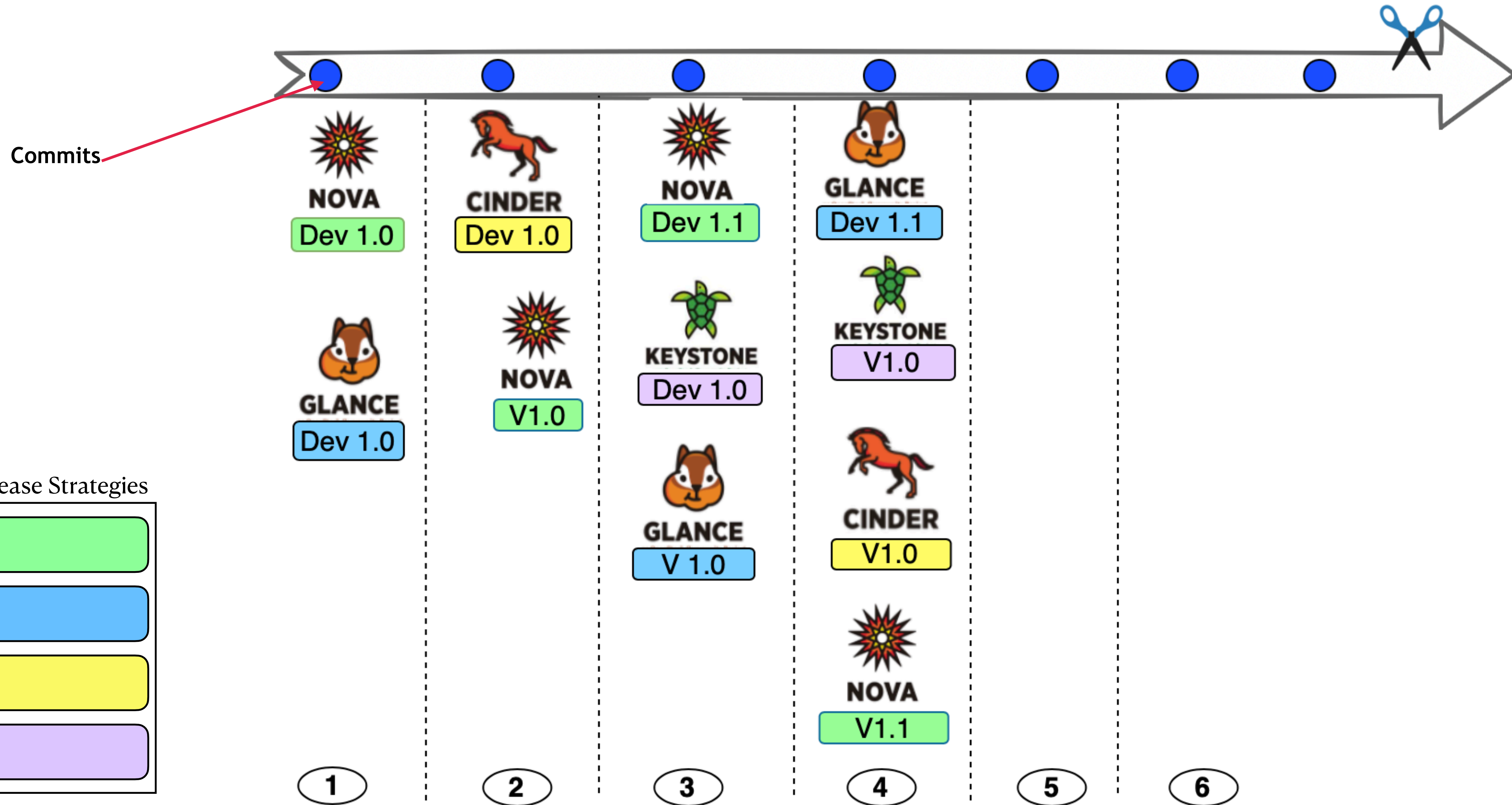
How to Release a SECO with each project having it's road-map?
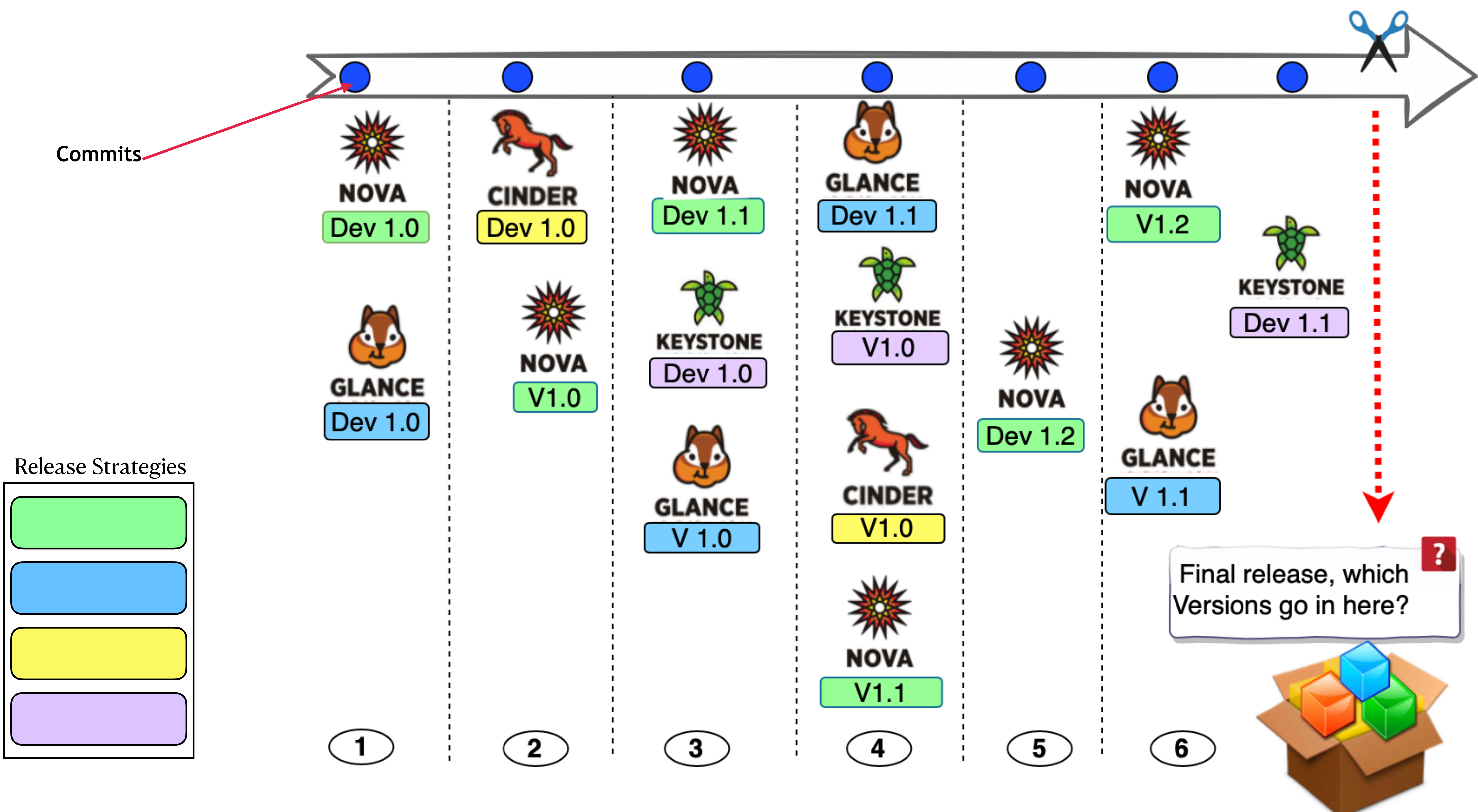
RQ: *What does it take for an ecosystem release team to manage synchronized releases across time successfully?*

Our research question aims at Identifying the **"What"**, **"How"** and **"Why"**
of the release synchronization process at SECOs' level

Details available in our paper online at:

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

We apply Qualitative analysis on IRC chat logs, follow up with Semi-structured interview and Mined OpenStack code-base repositories
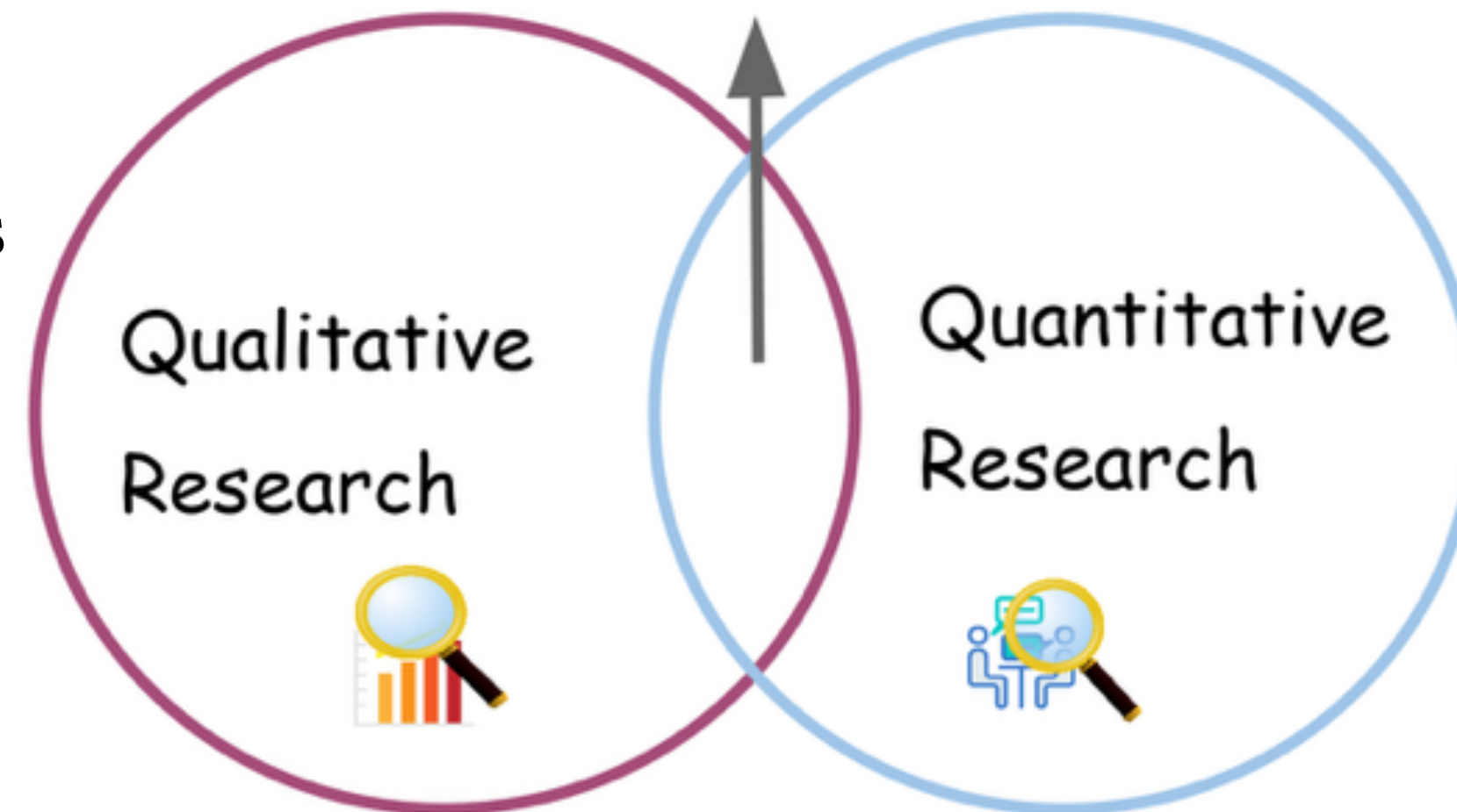

(1) Qualitative Analysis of 52 weeks of IRC logs

(2) Open-ended Interview with 8 experts

(3) Mined over 2500 repos coordinated by a central release team

Emerging themes from labeling qualitative data, which resulted to release sync activities by the release team

Excerpt of an Affinity diagram showing 3/10 release Sync. Activities

Automation/Tooling Management		Release Model Management		Release Planning/Tracking	
Tool Development	Script to automate tasks	Projects Release Model	Cycle with Milestones	PTG scheduling	Release cycles planning
Change of workflow	Replace deprecated tools	Cycle with release candidates	Projects following Cycle with intermediary	Release cycle timeline	First meeting of release cycle
semi-automatic validation of work	Building tools to automate processes	Tagging cycle-with-milestone projects	Release Project Model and Milestone	Scheduling tasks and Events	Release cycle planning

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

We validated our findings with the release team experts for Completeness and Accuracy

We missed one activity, **Cycle Highlight Management**, which was introduced after the studied Queens release.

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

Quantitatively, we mined and analyze over 2.5K repos managed by the release team to understand the co-existence of different release strategies.
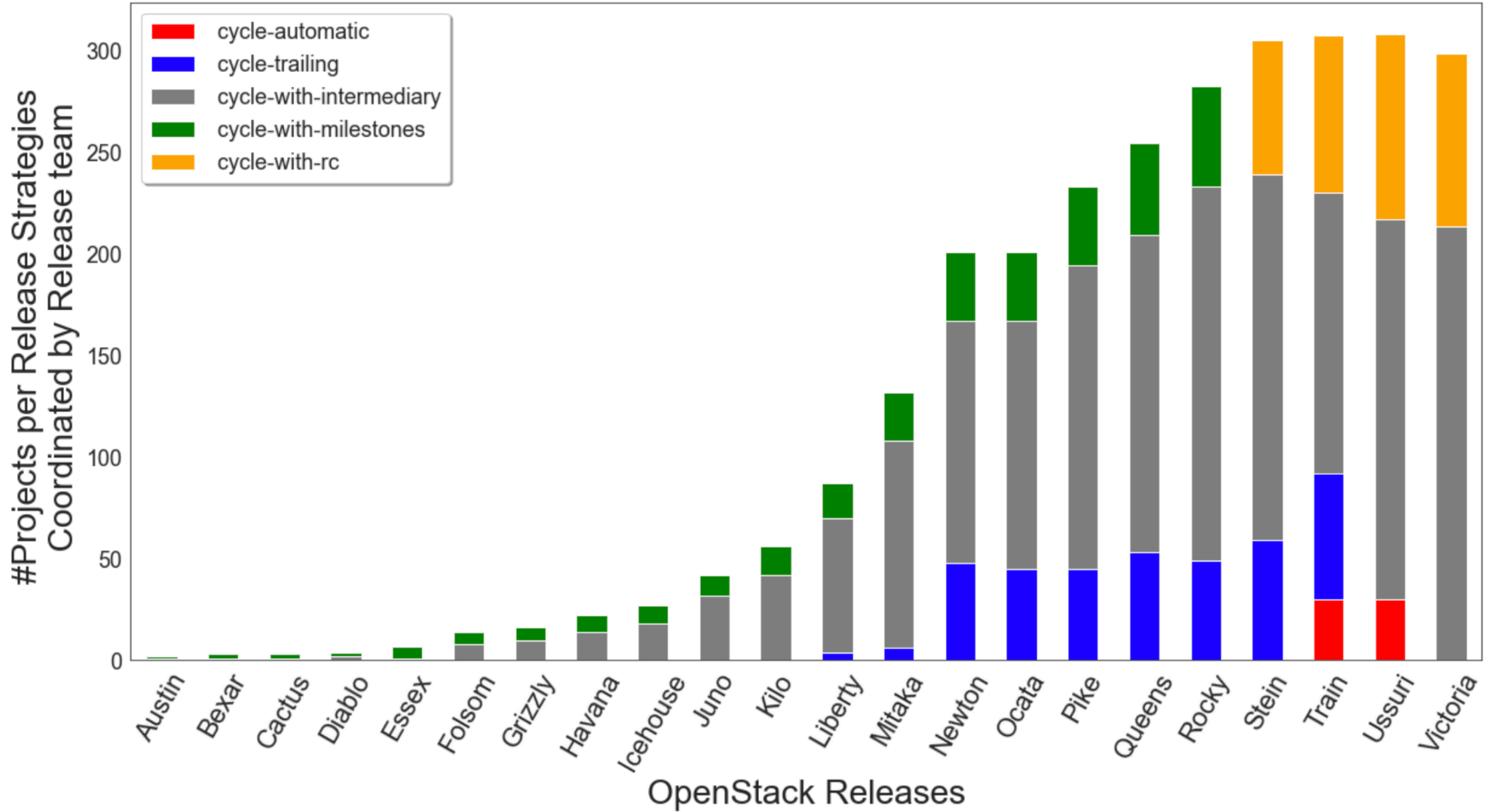
1. Cycle-with-intermediary (1746)

2. Cycle-with-rc (319)

_____

3. Cycle-with-milestones (308)

4. Cycle-trailing (371)

5. Cycle-automatic (60)

The Evolution of release strategies used by OpenStack projects (#)

The Evolution of release strategies used by OpenStack projects (#)

Template used to capture the catalog of 10 Release Sync. Activities

What Brief outline of the goal of the activity.

Why Short description of the rationale behind the activity.

How The major tasks involved in the activity based on the OpenStack observations.

Example Illustration(s) of discussions about this activity in the context of OpenStack.

Expert Insights and feedback provided by the interviewed OpenStack experts.
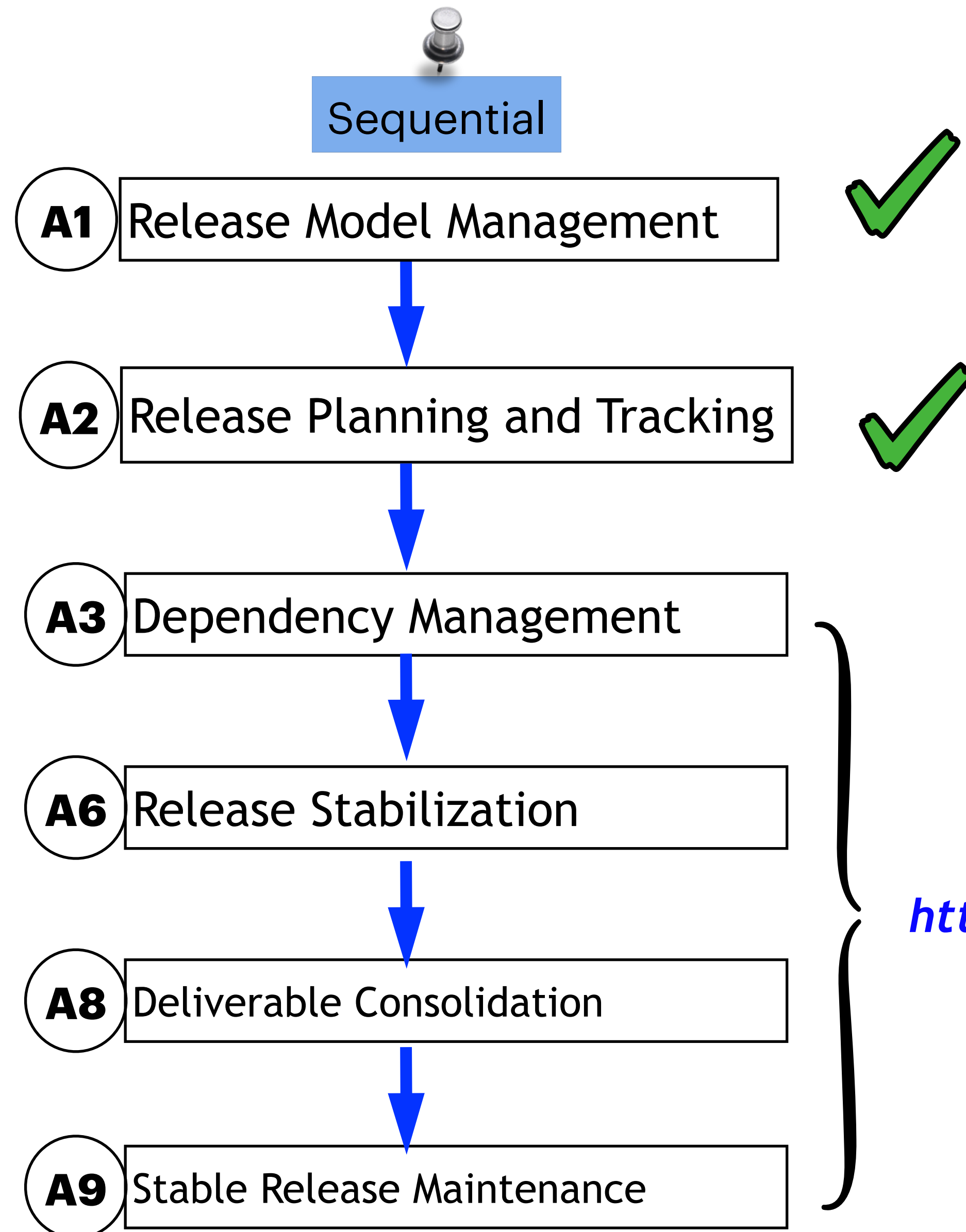
Release sync. Socio-technical Activities performed by the Centralized release team.

Socio-Technical Activities

ID	Release synchronization activity	%Mitaka	%Ocata	%Queens
A1	Release Model Management	48	46	44
A2	Release Planning and Tracking	100	100	100
A3	Dependency Management	84	70	72
A4	Establishing a Network of Trusted Liaisons	36	34	40
A5	Establishing Communication Channels	48	40	40
A6	Release Stabilization	52	53	56
A7	Tools/Automation Management	80	83	84
A8	Deliverable Consolidation	76	75	80
A9	Stable Release Maintenance	40	49	48
A10	Cycle Highlights Management	N/A	N/A	N/A

Dependencies among nine release socio-technical activities

Sequential

A1 Release Model Management ✓

A2 Release Planning and Tracking ✓

A3 Dependency Management

A6 Release Stabilization

A8 Deliverable Consolidation

A9 Stable Release Maintenance

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

17

# Dependencies among nine release socio-technical activities

Sequential

| A1 | Release Model Management |

| A2 | Release Planning and Tracking |

| A3 | Dependency Management |

| A6 | Release Stabilization |

| A8 | Deliverable Consolidation |

| A9 | Stable Release Maintenance |

Social

| A4 | Establishing Trusted Liaisons | ✓

| A5 | Establish Communication Channels | ✓

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

Dependencies among nine release socio-technical activities
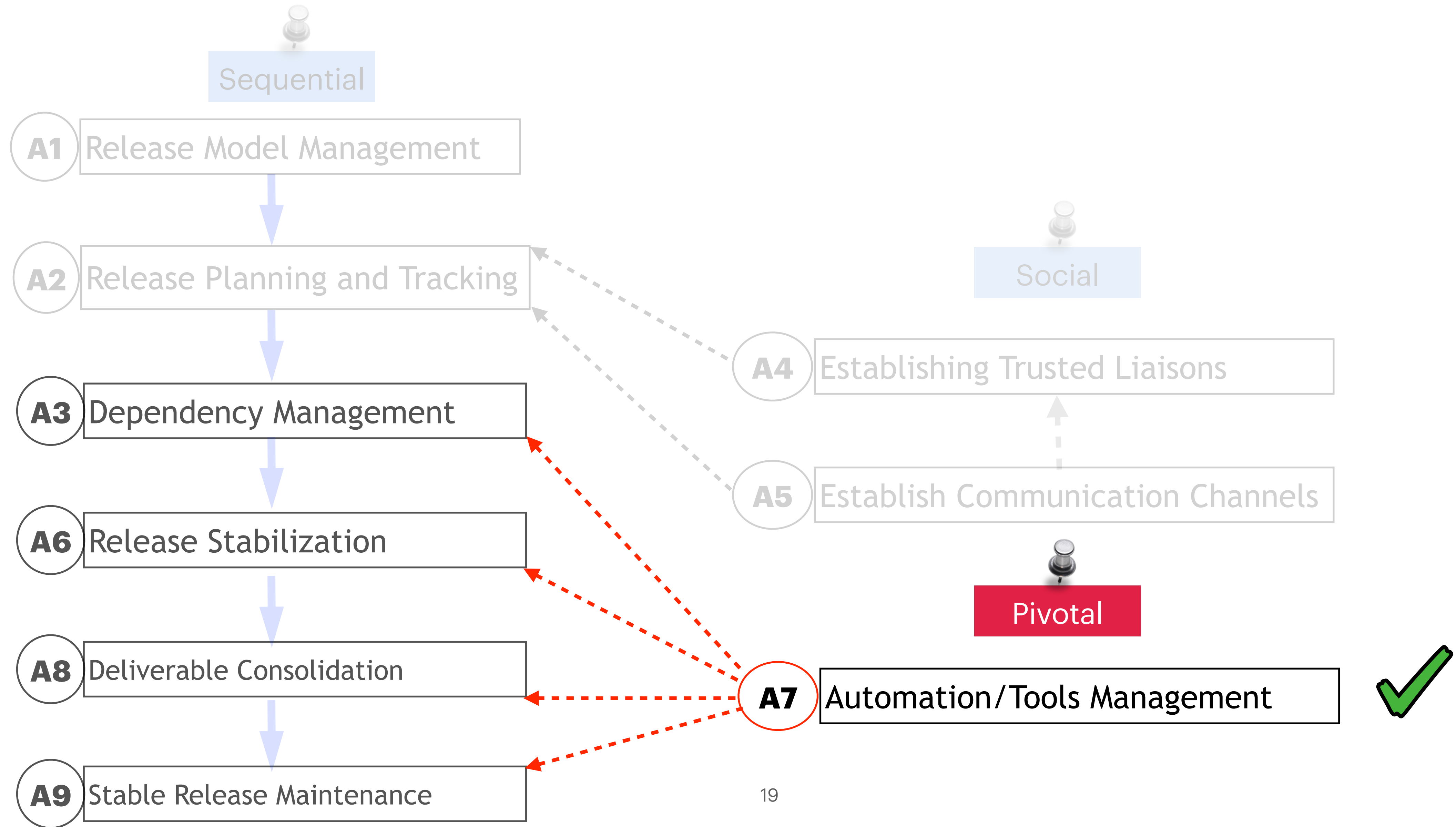
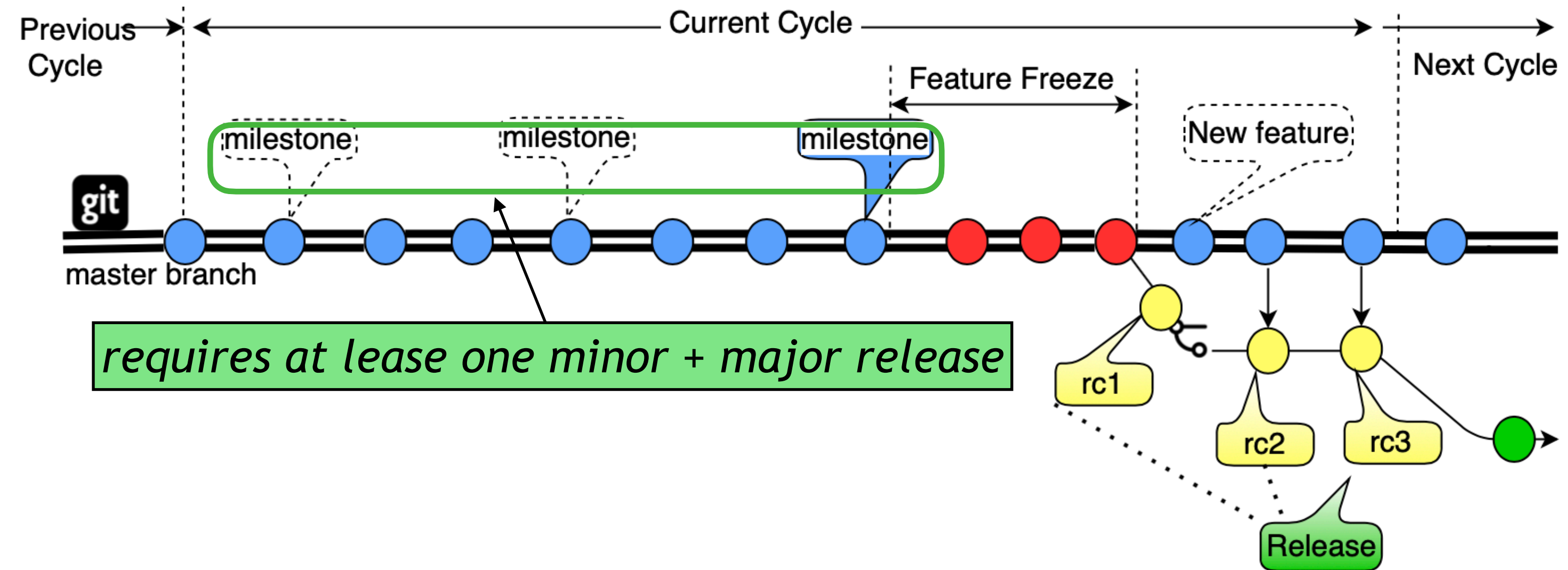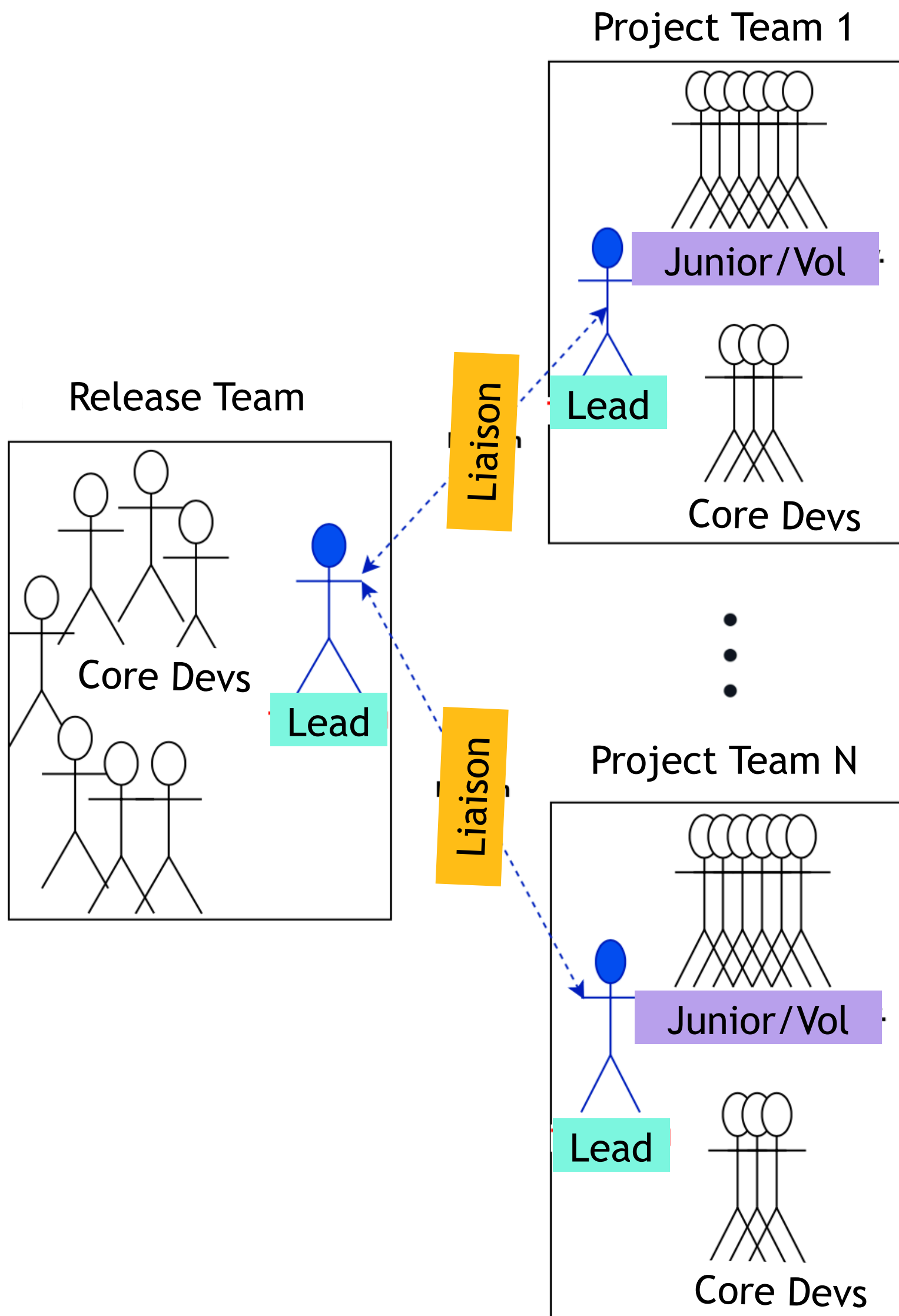# The Release Team Coordinates projects' Releases
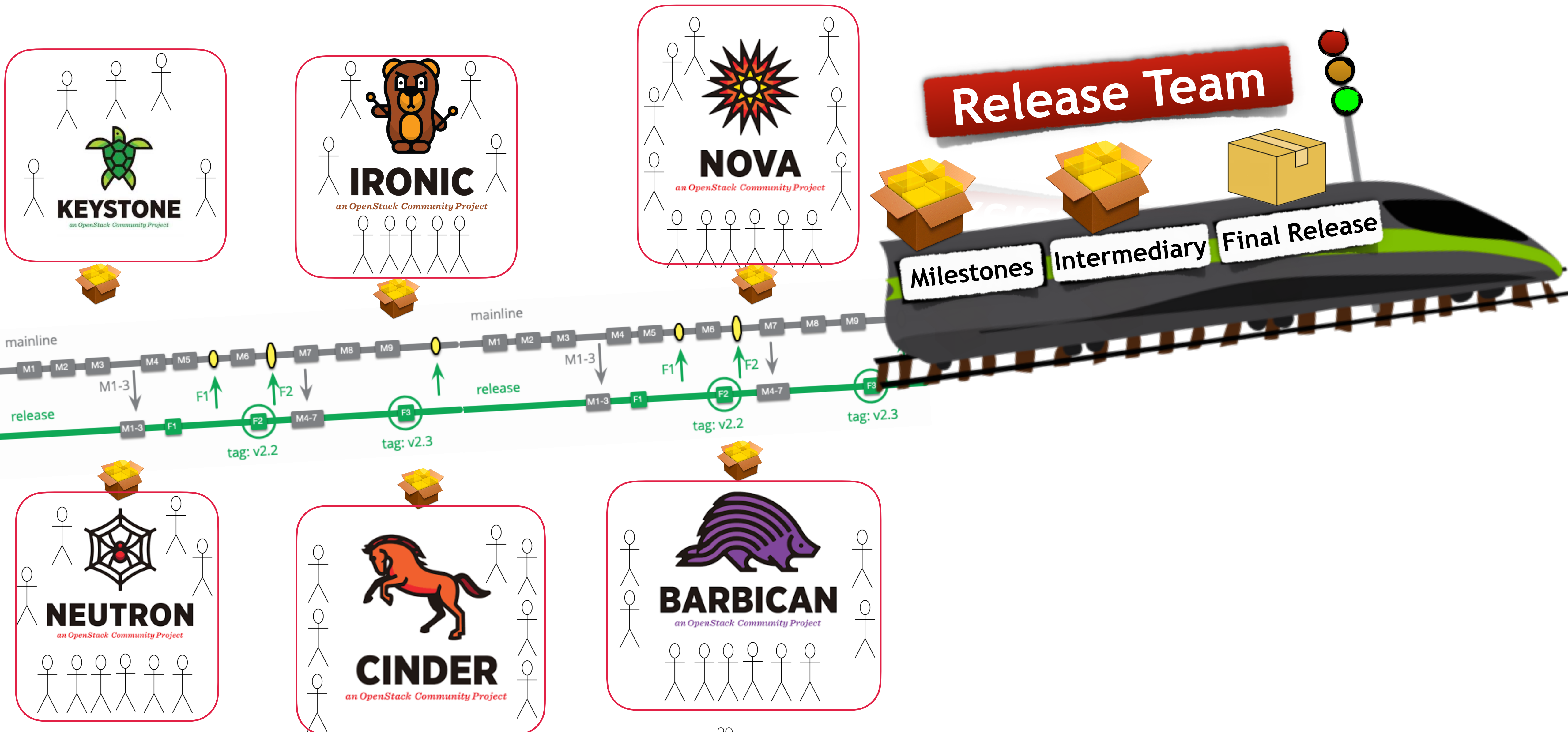
A Centralized release team coordinates the integration of each project release into final SECO release
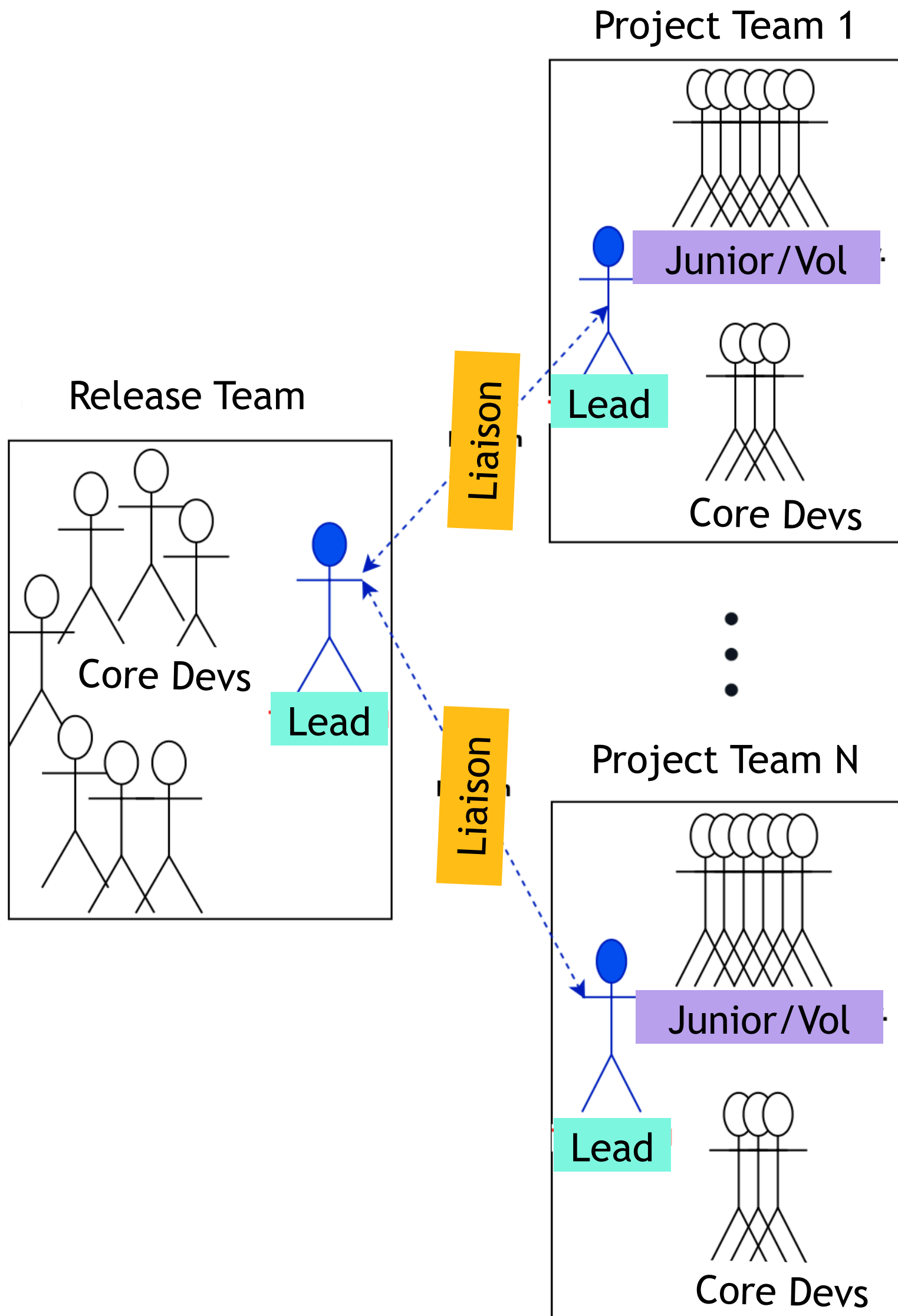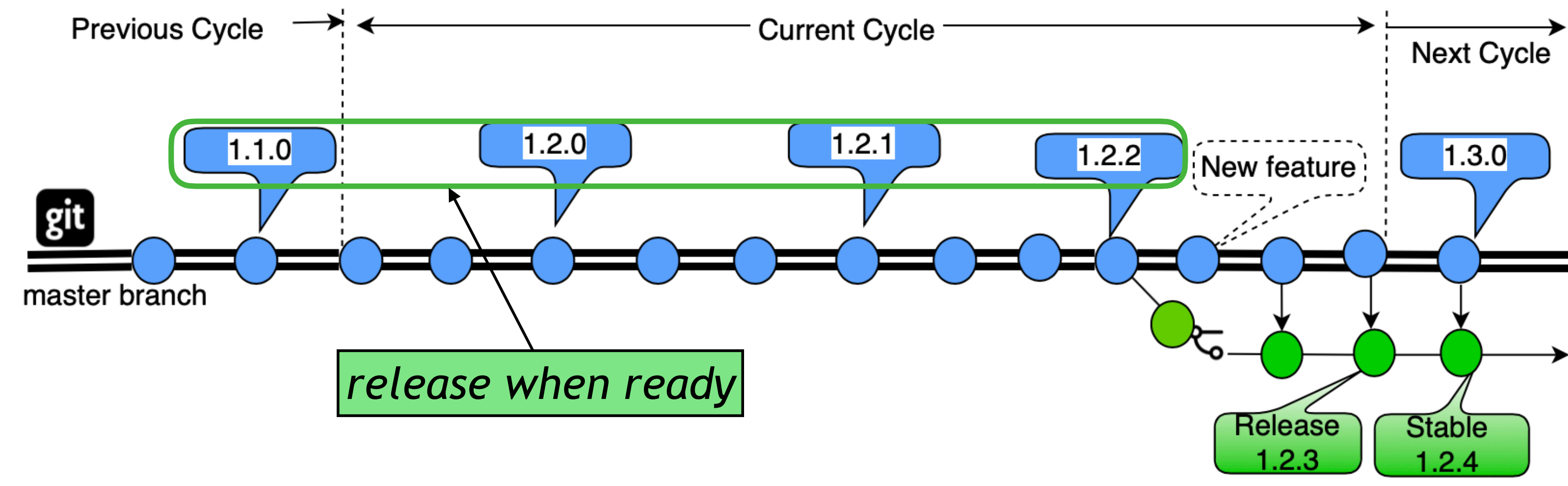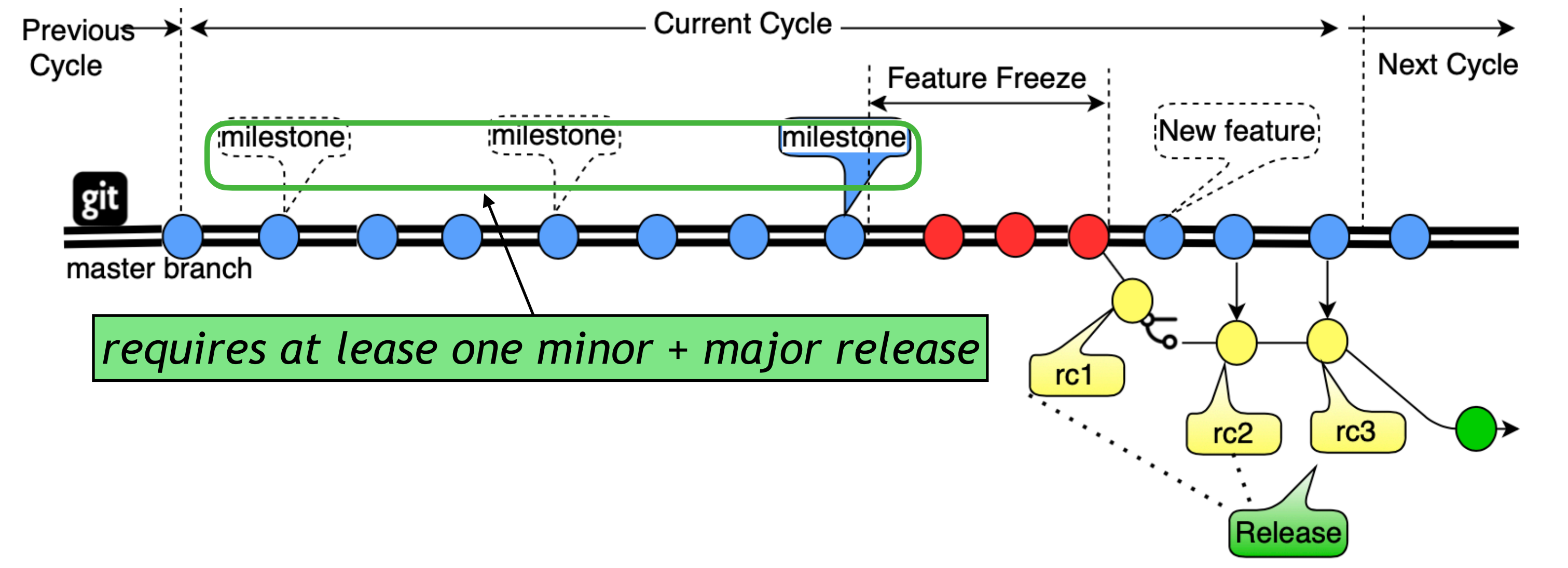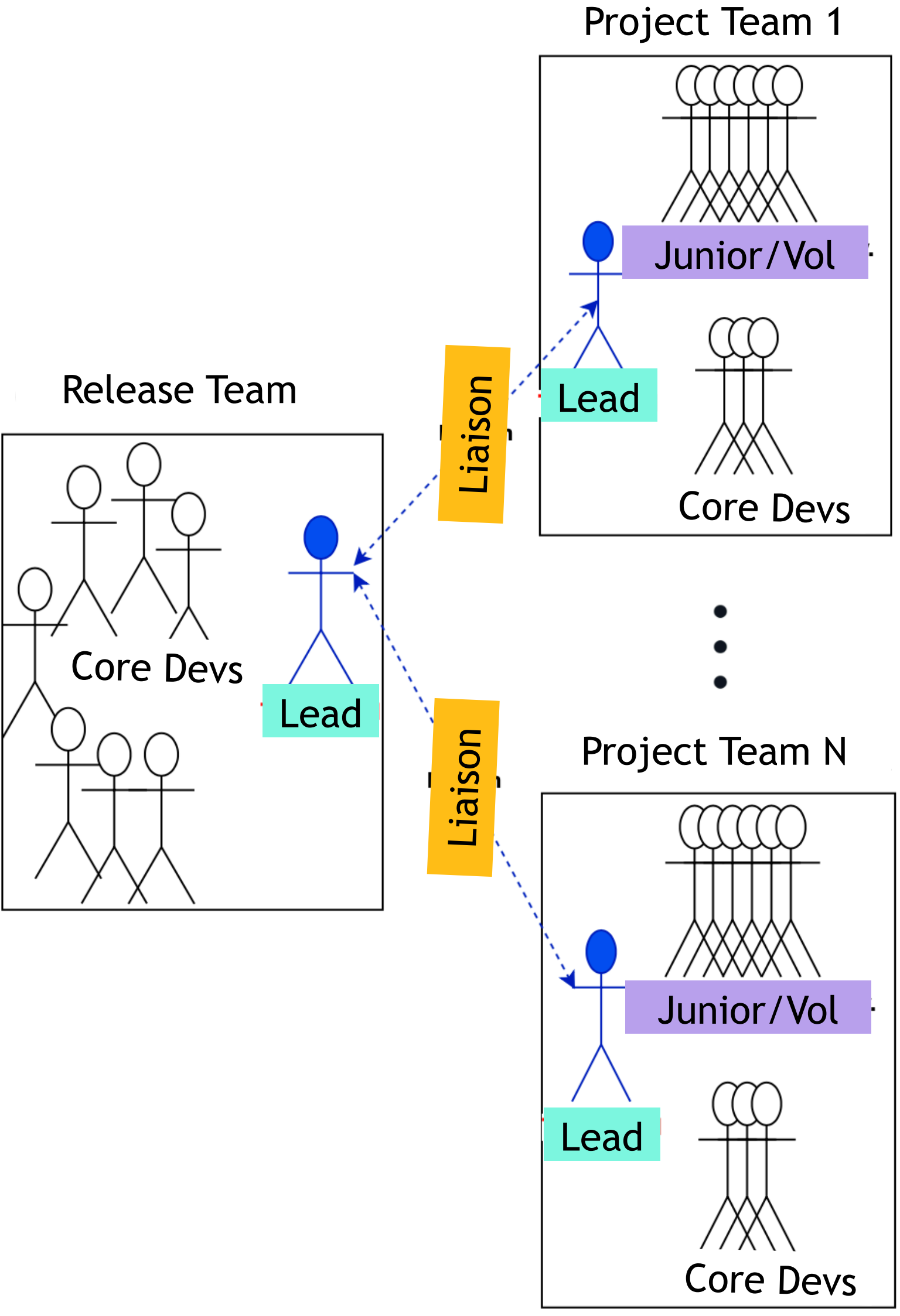
The Centralized release team manage multiple strategies to coordinates each project release to a final SECO release

The Centralized release team manage multiple strategies to coordinates each project release to a final SECO release

Project Team 1

Release Team

Core Devs

Lead

Liaison

Junior/Vol

Lead

Core Devs

Project Team N

Junior/Vol

Lead

Core Devs

Previous Cycle

Current Cycle

Feature Freeze

Next Cycle

New feature

milestone

milestone

milestone

git

master branch

requires at lease one minor + major release

rc1

rc2

rc3

Release

Cinder V1.0

Previous Cycle

Current Cycle

Next Cycle

New feature

1.1.0

1.2.0

1.2.1

1.2.2

1.3.0

git

master branch

release when ready

Release 1.2.3

Stable 1.2.4

Glance V1.1

We used time-series analysis to generalize our findings and found similar patterns of release synchronization activities in both Eclipse and Gnome ecosystems
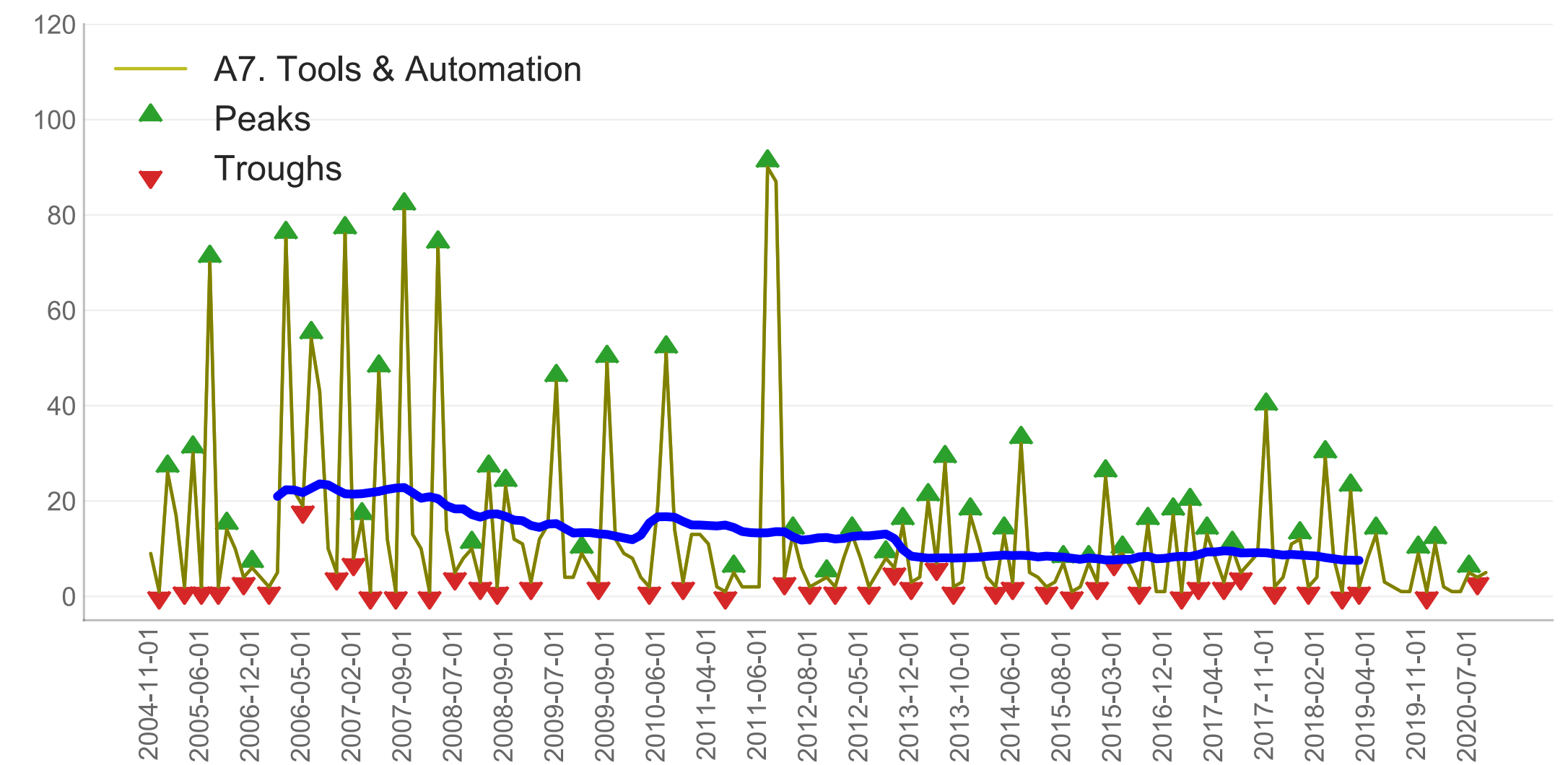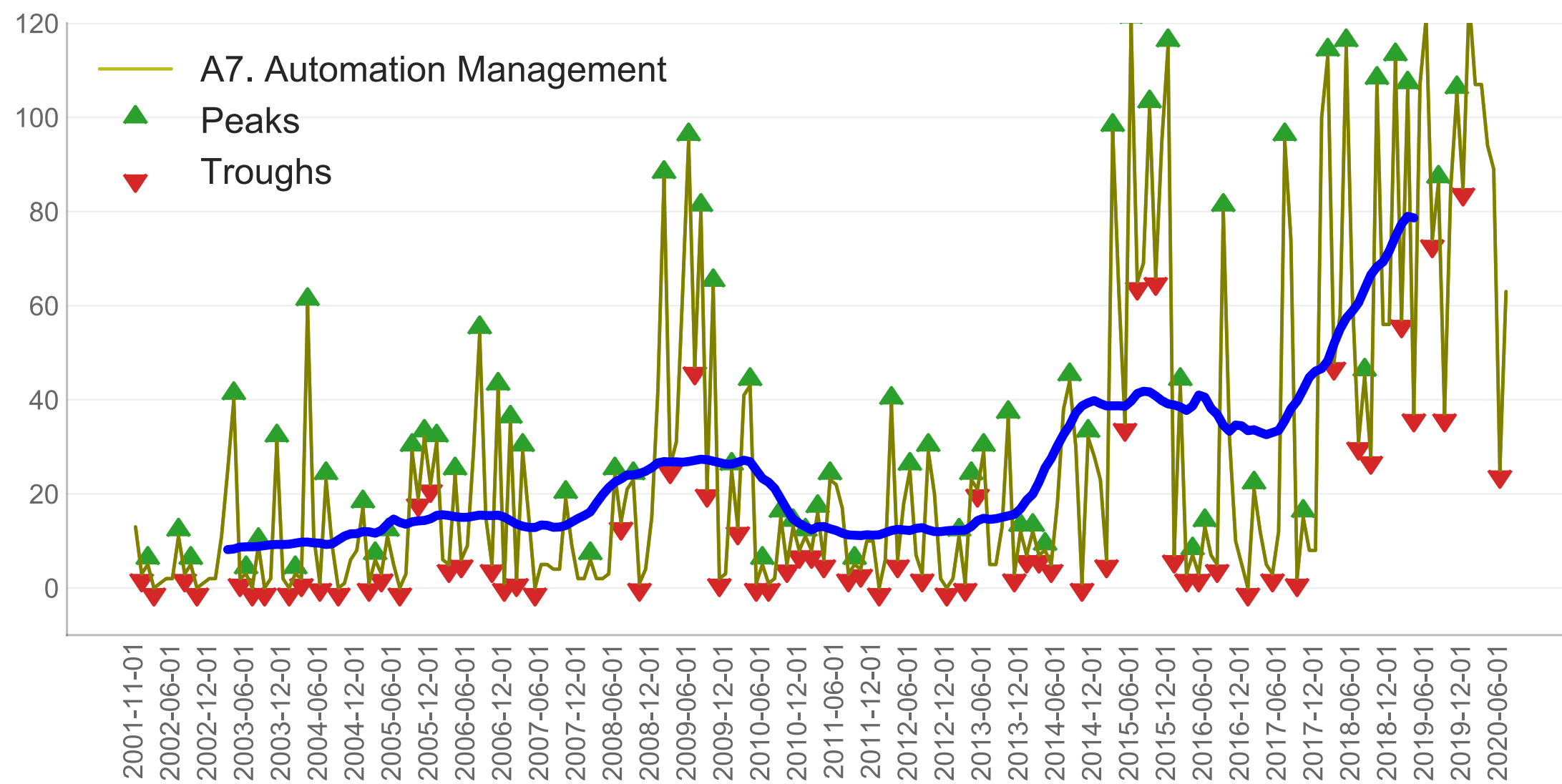
Persona of the Release team showing key skills and Challenges to become a Release engineer

Release Engineer

Bob Stevino | 28 yrs

Location: Online team

BSc. Software Engineering, 2005.

Bio

openstack.

Bob got hired at OpenStack in 2010, formerly from Rackspace and NASA. He is skilled in scripting and functional languages, such as Python.

Exciting Moment

Bob's pleasant moment is when he helps project teams get their deliverable out timely.

Summary

" *Bob coordinates the releases of one of the world's most massive and fast-growing ecosystem in cloud computing; OpenStack.* "

Daily tasks

Bob coordinates project teams on key activities.
Bob is willing to pick up tasks when no one else is ready to do the job.
Besides, Bob manages the automation around the release pipeline.

#irc chat @bob

@ bob.stev@OpenStack.org

Hobby

- Hacking   - Swimming
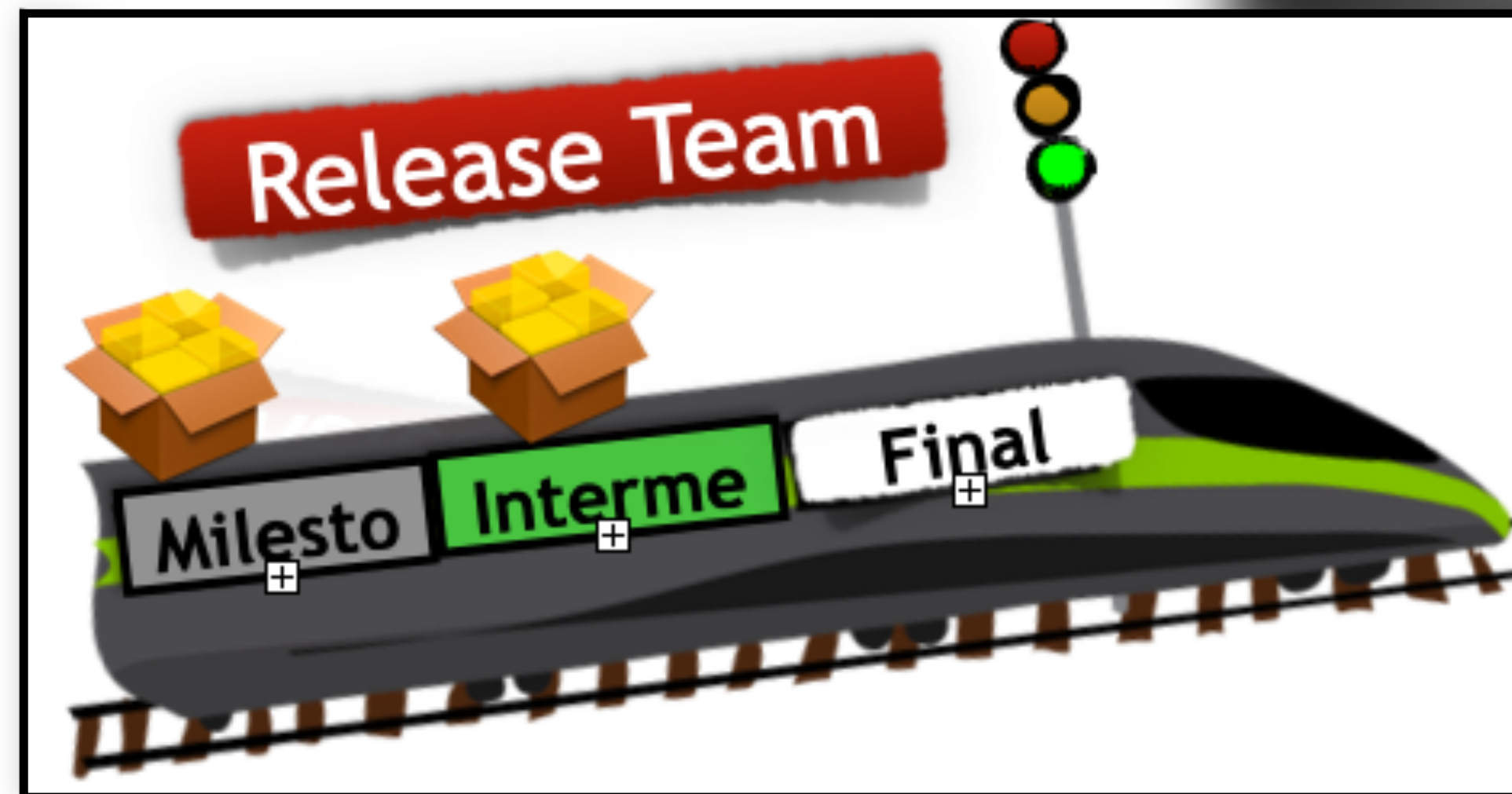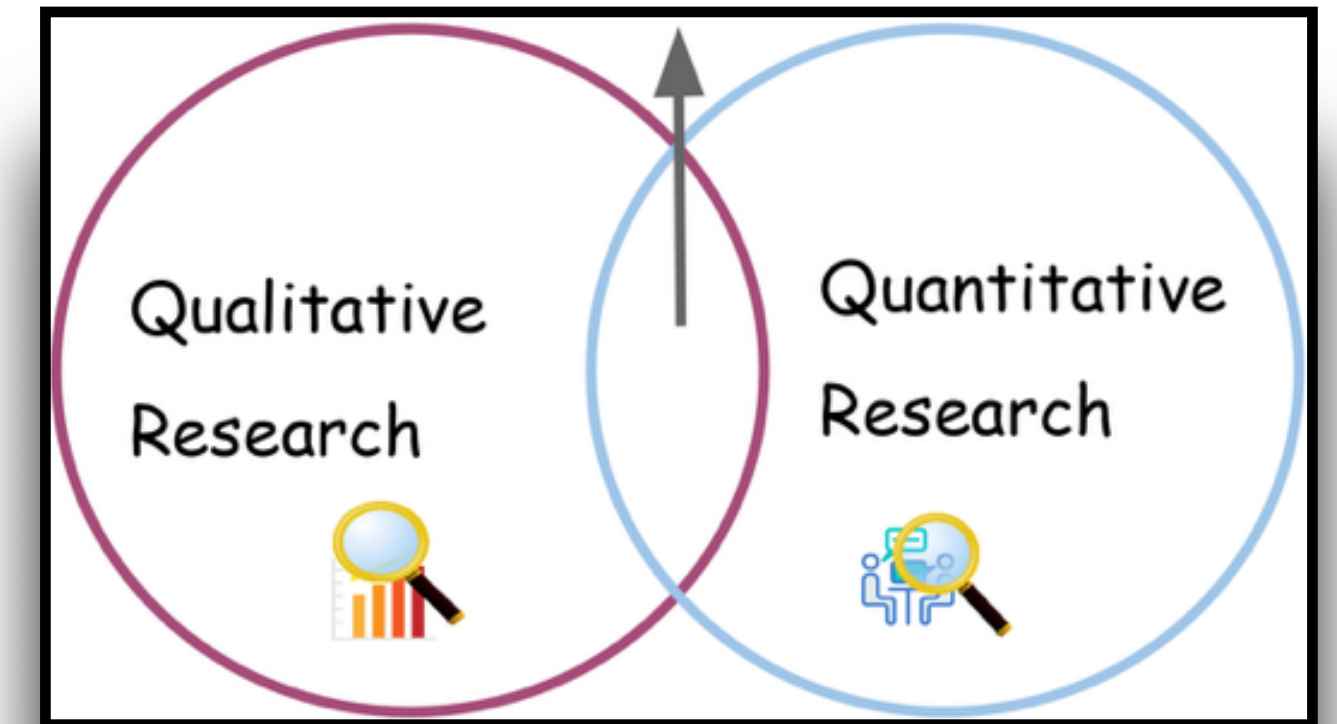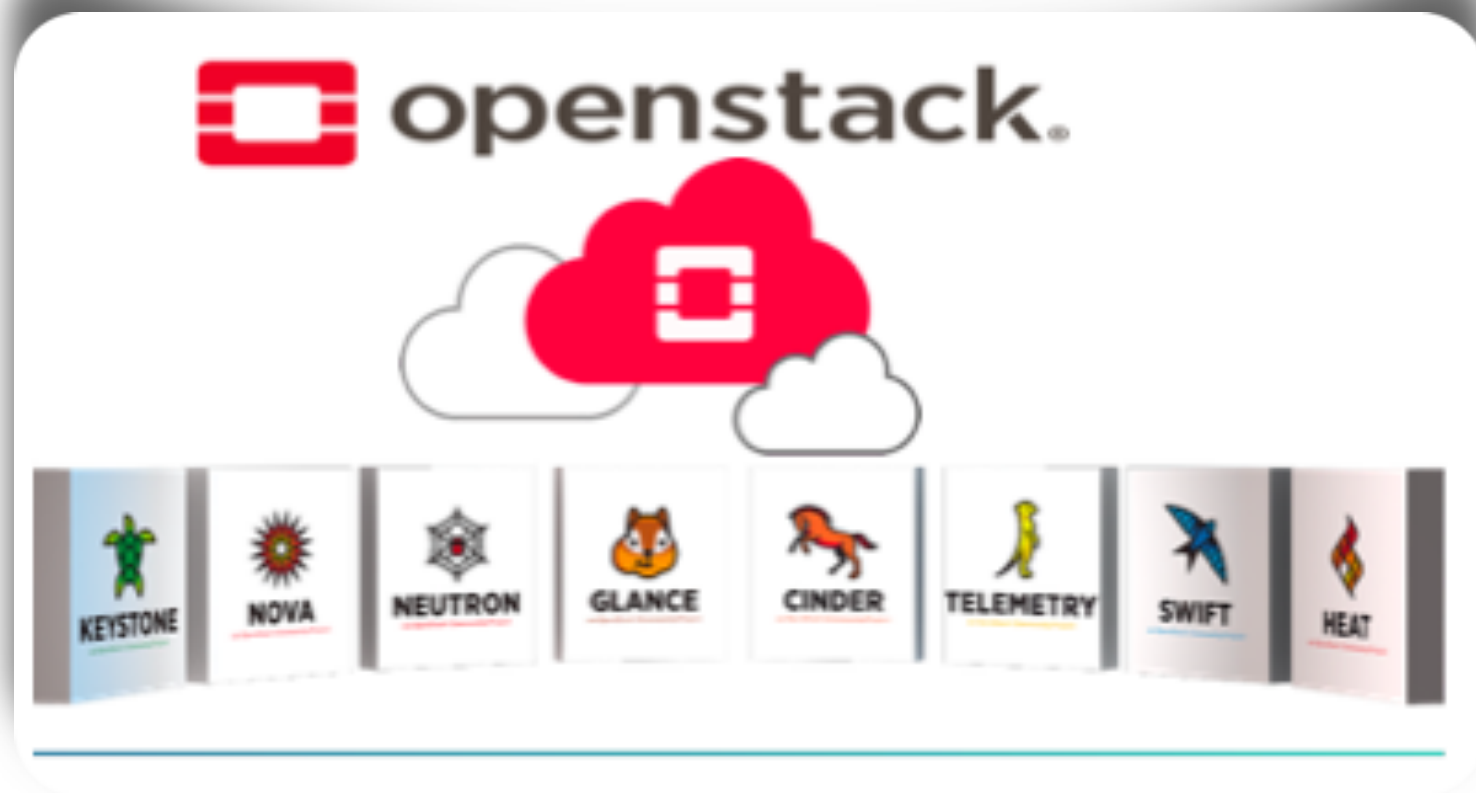- Site seeing   - Cooking
- Tennis   - Dancing
- Playing Chess

Key Activities

1. Release model Mgt	2. Planning and Tracking	3. Dependency Mgt	4. Trusted Liaisons	5. Communication
6. Release Stabilization	7. Automation	8. Delivery Consolidation	9. Stable Maintenance	10. Cycle Highlights

Take home message

https://mcis.cs.queensu.ca/publications/2021/emse_armstrong.pdf

a.foundjem@queensu.ca