

Building and Managing an Open Source Program Office

Ibrahim Haddad, Ph.D.
Vice President, Strategic Programs
Linux Foundation



@IbrahimAtLinux



IbrahimAtLinux.com



linkedin.com/in/ibrahimhaddad/

Agenda

- Criticality of open source
 - Introduction to OSPOs
 - OSS transformation
 - Organizational models
 - Staffing
 - Roles and responsibilities
 - Challenges
 - Recommendations practices
 - Conclusion
-

Criticality of open source software

Strategic importance of OSS

- 1 Neutral environment for collaboration & cross-pollination
- 2 Innovation multiplier - community driven
- 3 Minimizes fragmentation and supports the upstream development model
- 4 Enables better interoperability
- 5 Facilitates standardization on open technologies
- 6 Qualifies reference architectures
- 7 Lowers barriers to enter a new domain
- 8 Enables business opportunities supported by a flexible licensing model
- 9 Leads to better products, improved quality and security
- 10 Allows fast trailing twelve months and shared cost of development

Strategic impact of OSS on the organization



- Accelerate the development of open solutions
- Provide an implementation to an open standard



- Commoditize a market
- Reduce process of non-strategic software assets
- Provide an implementation to an open standard
- Share development costs



- Drive demand by building an ecosystem for products and services



- Partner with others
- Engage customers
- Strengthen relationships with common goals

OSS enables products and services

How Open
Source
Enhances
Enterprise
Products

Open Source Direct Product Enablement

- Implement open source development requests from R&D and product teams
- Contributing internal code into open source projects
- Support open source compliance efforts in the organization



Open Source Indirect Product Enablement

- Stabilize upstream projects
- Participate in policy discussions
- Influence upstream projects via thought leadership and code contributions
- Participate in upstream technical discussions



Upstream Development Enables Better Products

- Less work for product teams
- Minimized maintenance cost of source code
- Increased code quality
- Faster development cycles
- More stable code bases
- Improved reputation in upstream projects

OSS improves talent and skills development

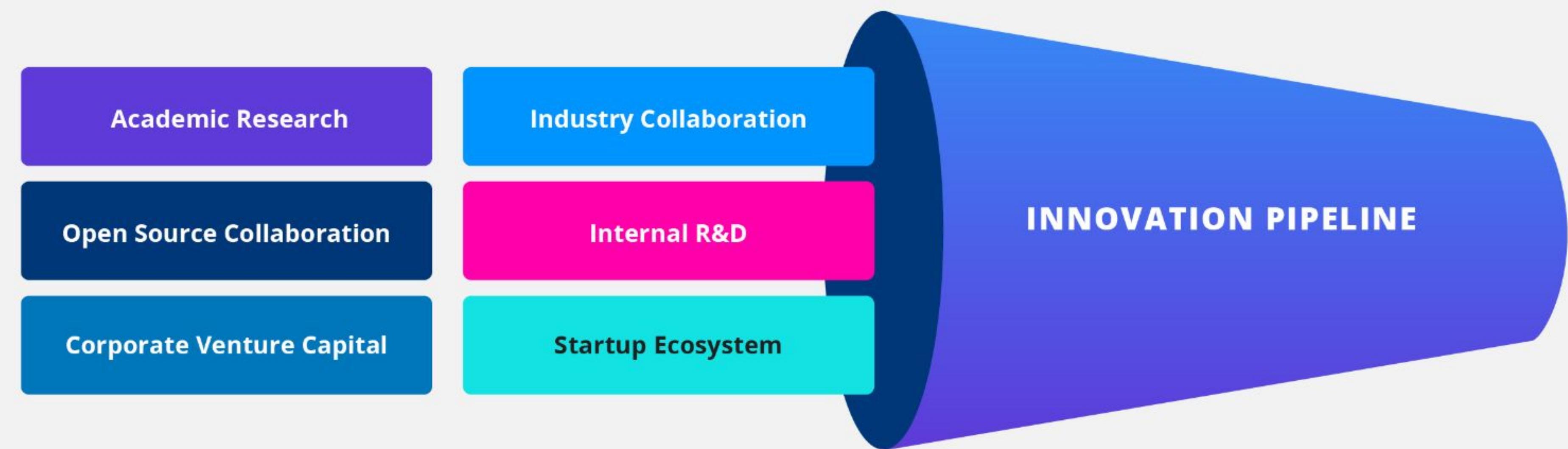
OPEN SOURCE OPPORTUNITIES

OSS can improve enterprise product development and
IMPROVE ENTERPRISE TALENT AND SKILLS DEVELOPMENT.

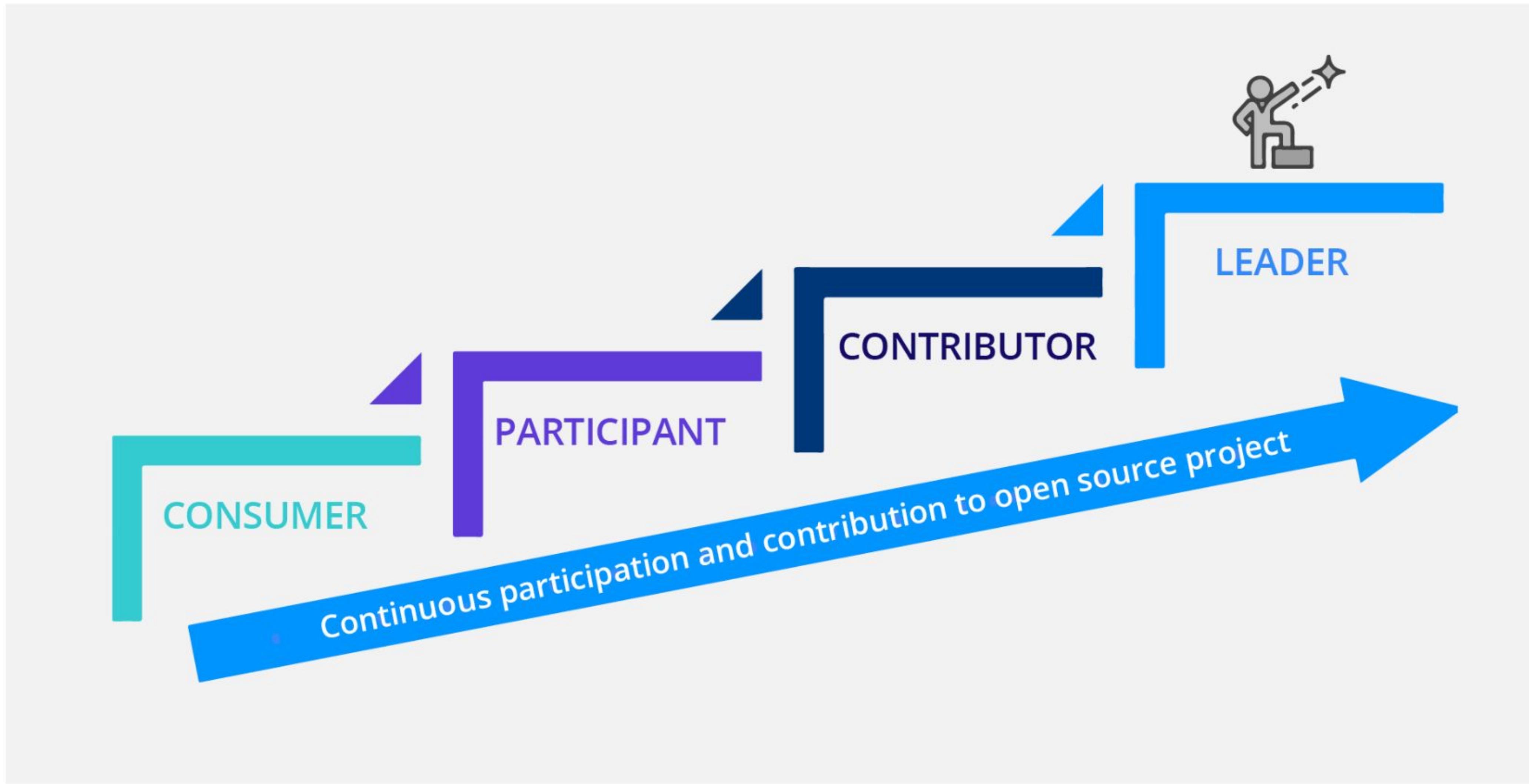


OSS is a building block of the innovation pipeline

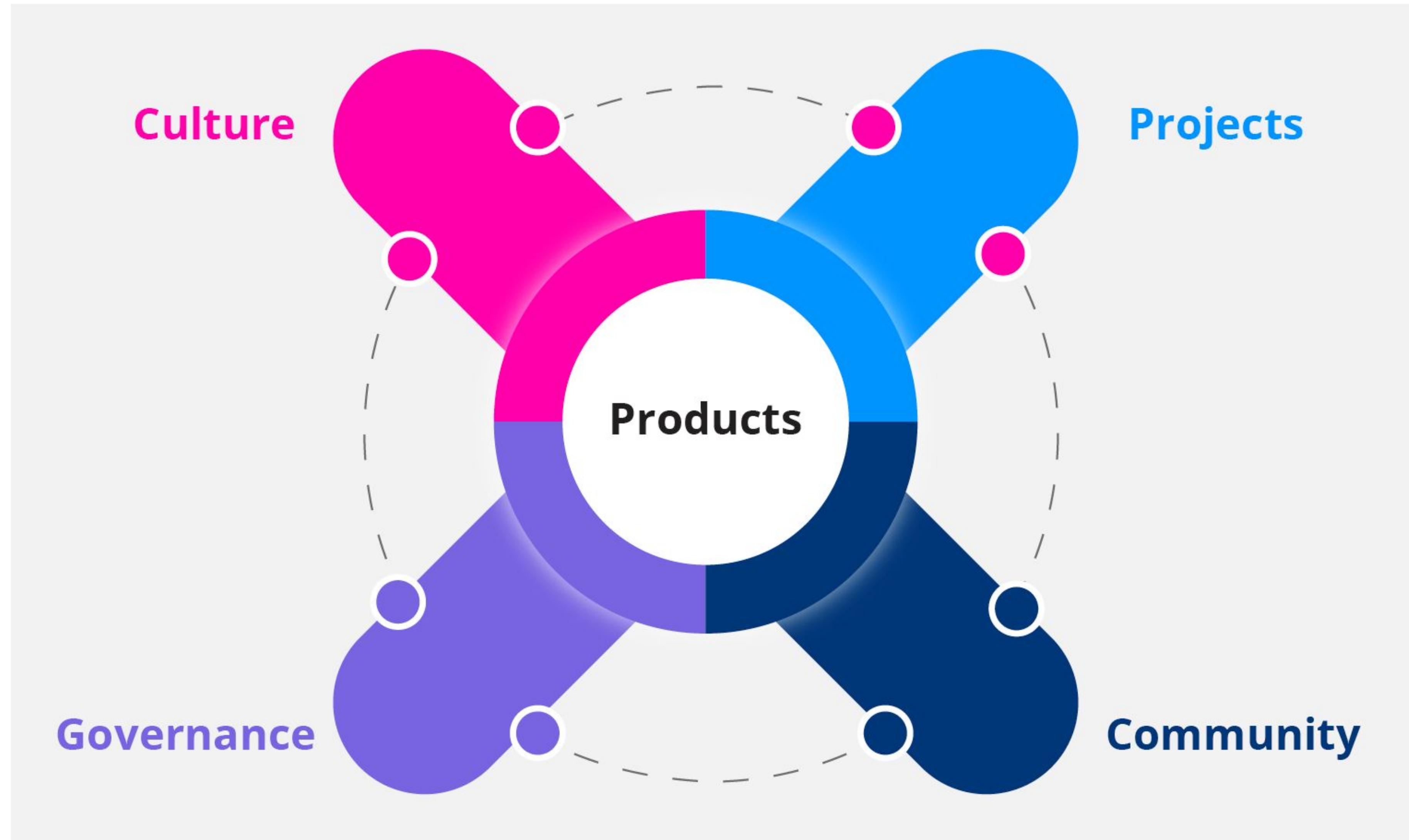
Open Source
Accelerates Product
Innovation



Stages of an Enterprise Open Source Journey



The Four Poles of an Open Source Strategy



Core Elements in an Enterprise Open Source Infrastructure

Community

Contribution

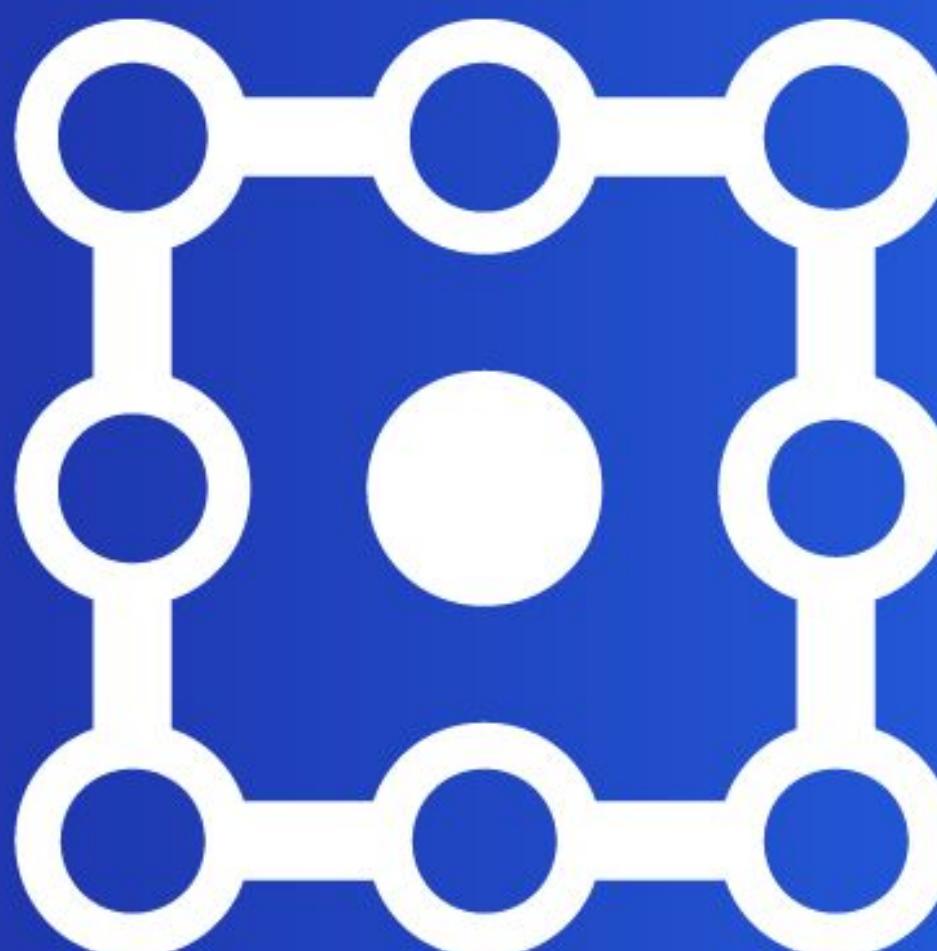
Compliance

Consumption

Introduction to OSPOs

WHAT IS AN OSPO?

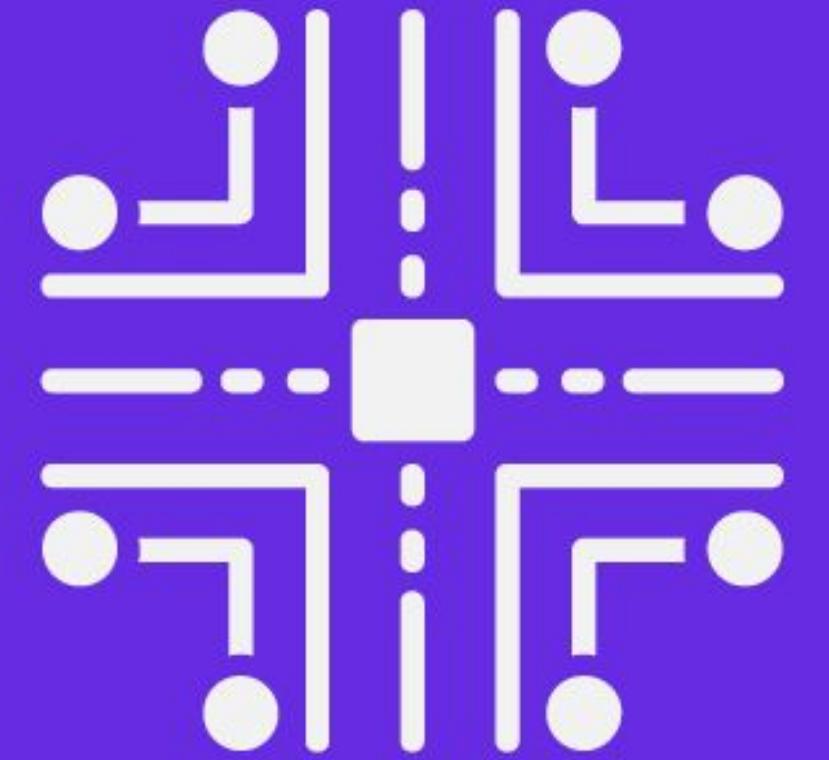
OSPO is designed to be **the center of competency for an organization's open source operations and structure.**



OSPOs manage all open source activities across the organization including consumption, compliance, contribution, communication and community related activities.

ENTERPRISE INFRASTRUCTURE

Set up your organization to implement open source, from compliance to consumption to contribution.



Why organizations should consider setting up an OSPO?



WHY FORM AN OSPO?

Organizations across industries establish OSPOs to **drive OSS leadership and gain a critical foothold in a robust, external R&D ecosystem.**



Adoption of OSPOs or
similar programs has risen
to 50% of respondents,
**THE HIGHEST LEVEL
IN 5 YEARS.**

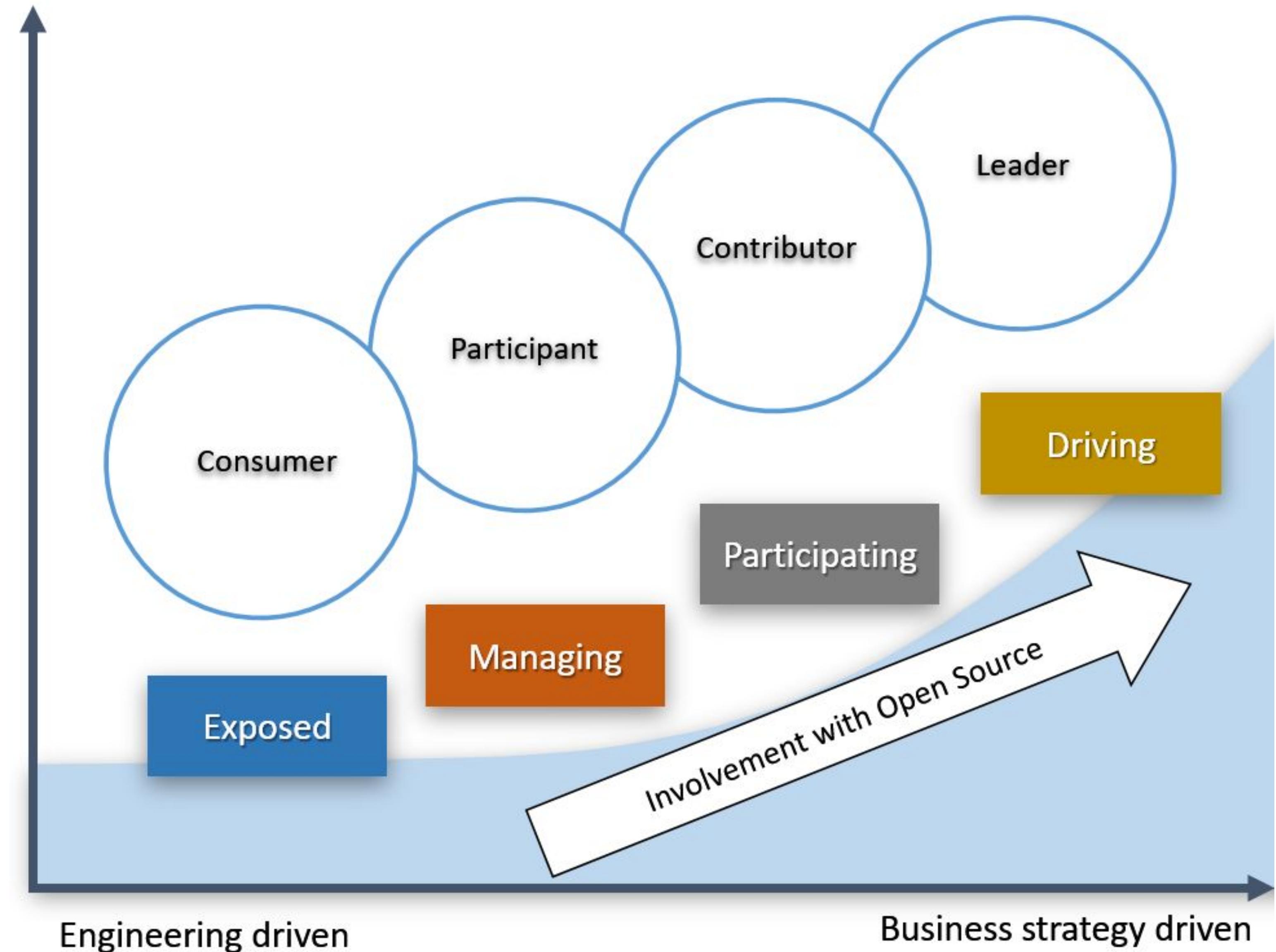
OSPO GROWTH

TODO Group - 2022 Survey
<https://github.com/todogroup/osposurvey/tree/main/2022>

OSS & Digital Transformations

**Where should
you start?**

**Identify your current
and target positions,
then chart your path.**



Consuming OSS

Aggressively consuming open source components will increase your ability to differentiate and reduce overall time and cost to deliver products.

- Approach for an open source consumption strategy:
 - Use a strategic classification scheme to guide decisions on what open source software to consume
 - Ensure the company meets all license obligations of its use of open source software
 - Deploy automated workflows for evaluating/approving open source usage
 - Establish an Open Source Review Board (OSRB) to serve as a clearinghouse for all open source activities
 - Invest in headcount and infrastructure for engineering, product management, and legal to manage a complex mix of closed source / open source software

Consumption and Compliance Infra

Strategy	Portals	Policy & Process	Development	Team	Education	Tools	Linux Foundation
Compliance	Internal site (educational)	Usage and compliance policy	Integrate compliance in development and QA process	Compliance team (core and support)	Training on company policy	Source code scanning	OpenChain
Managing inquiries	External site (obligation fulfillment, source code distribution)	Distribution Auditing Notices Usage	Integrate compliance tools with build systems and developer workflow	Scoreboard and success metrics	Guidelines and best practices	Linkage analysis	Software Package Data Exchange
Licensing and risk tolerance	Internal messaging	Obligation fulfillment			Training on open source licenses	Dependencies analysis	Open Compliance Program
M&A and corporate development	External messaging	Mixing code under different licenses			New employee orientation	Security vulnerabilities analysis	TODO Group
Software procurement					Checklist for product team	Software bill of materials	Open Source Security Foundation
					Checklist for developers	Automation of online forms and workflow	
					Checklist for SW procurement	IP evaluation tool	
					Compliance mentorship	SW Inventory management	
					Professional formal training	Project management	

Compliance Challenges

- ACCESSIBILITY of the tool to all developers
- TRANSPARENCY of open issues and how they're being handled
- ADVANCED feature set to complement complex development
- SCALABLE supporting large code bases and large number of users
- SPEED to keep up with development pace
- ACCURACY with dependable repeating results

12 Rules for developers for better oss license compliance

Request approval to use open source software before you commit the code into the product repo

Request approval before linking proprietary code to an open source library or vice versa

Update the changelog of modified files to reflect the date of change, author, and a brief description of the change

Do not merge or mix source code incoming under different licenses without proper approval

Document interfaces between proprietary, third-party and open source software to help explain interaction and clarify any compliance concerns

Save an unaltered copy of the open source package in a backup location, along with its license information

When upgrading an open source component, verify if the license is still the same from the previous version

Do not remove or disturb existing licensing or copyright information. All such information must remain intact

Do not rename open source components

Do not copy/paste open source code (snippets) into proprietary or third-party code (or vice versa) without prior approval

Do not discuss compliance practices with individuals outside of your company

Always follow your company's policies and processes and when in doubt consult with your engineering manager

Contributing to OSS

- Selectively engage with targeted projects and communities to drive your company's needs.
- Contributing to strategic open source projects can help your organization gain additional value as code contributions shape feature development and steers the projects in a favorable direction to your products.
- Approach for an open source contribution strategy:
 - Incrementally invest in engineering, product management, and legal to engage with existing external communities.
 - Hire a staff director to lead open source strategy and manage the OSRB.
 - Hire contributors and committers to key open source communities that are critical to your products.
 - Deploy open source collaboration tools to support open source usage and contributions.

Contribution Infrastructure

Contribution	Dedicated Group	Open Standards
Policy & process on project contributions	Establish open source program office (OSPO)	Participate in relevant open standards
Guidelines & contribution training	Hire from open source projects	Consider open sourcing internal technology as reference implementation
Contribution approval team	Support & participate in open source foundations	
Increased participation in key open source projects	Host open source events	
	IT infrastructure to support open source development	
	Establish/recognize open source career path	
	Support communities of projects you depend on	

Open Source Contribution Infrastructure

How can you make it work?

- Create the right environment
- Provide the right incentives
- Provide the right exposure
- Integrate your teams within the specific projects' communities

BENEFITS OF UPSTREAMING CODE



Lower maintenance efforts for internally managed code, i.e., minimizes technical debt



Upstreamed code becomes visible to others and receives peer review and feedback, leading to improvements



Upstream contributions provide stability to the project. They send a signal that the project is useful and important, and that helps attract new contributors



Builds positive relationship between the contributing organization and the project community



Upstreaming code is an effective way to provide technical leadership and influence the project



Upstreaming contributes to easier compliance, improved security due to centralizing code in upstream repos

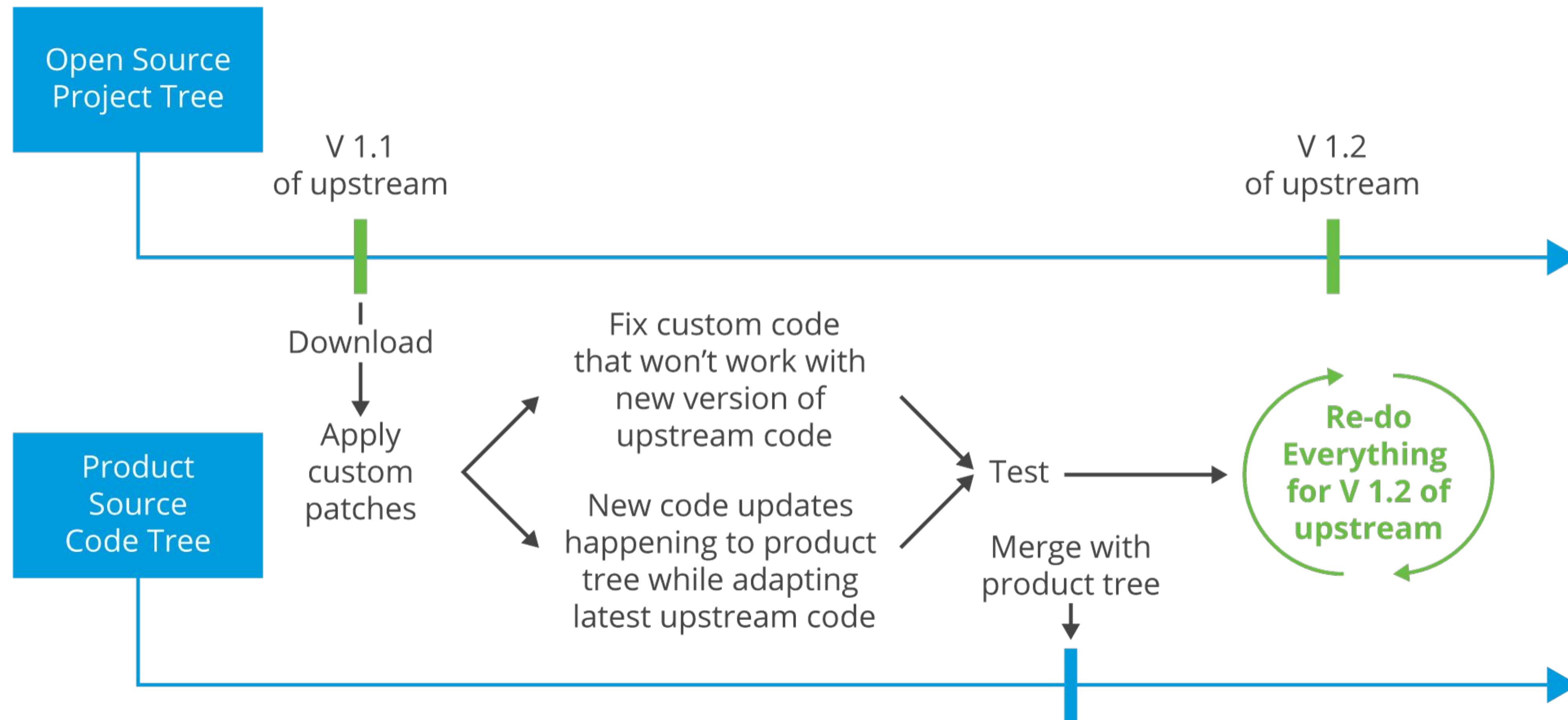


Upstream contributions are an effective means of ensuring stability in a company's software supply chain

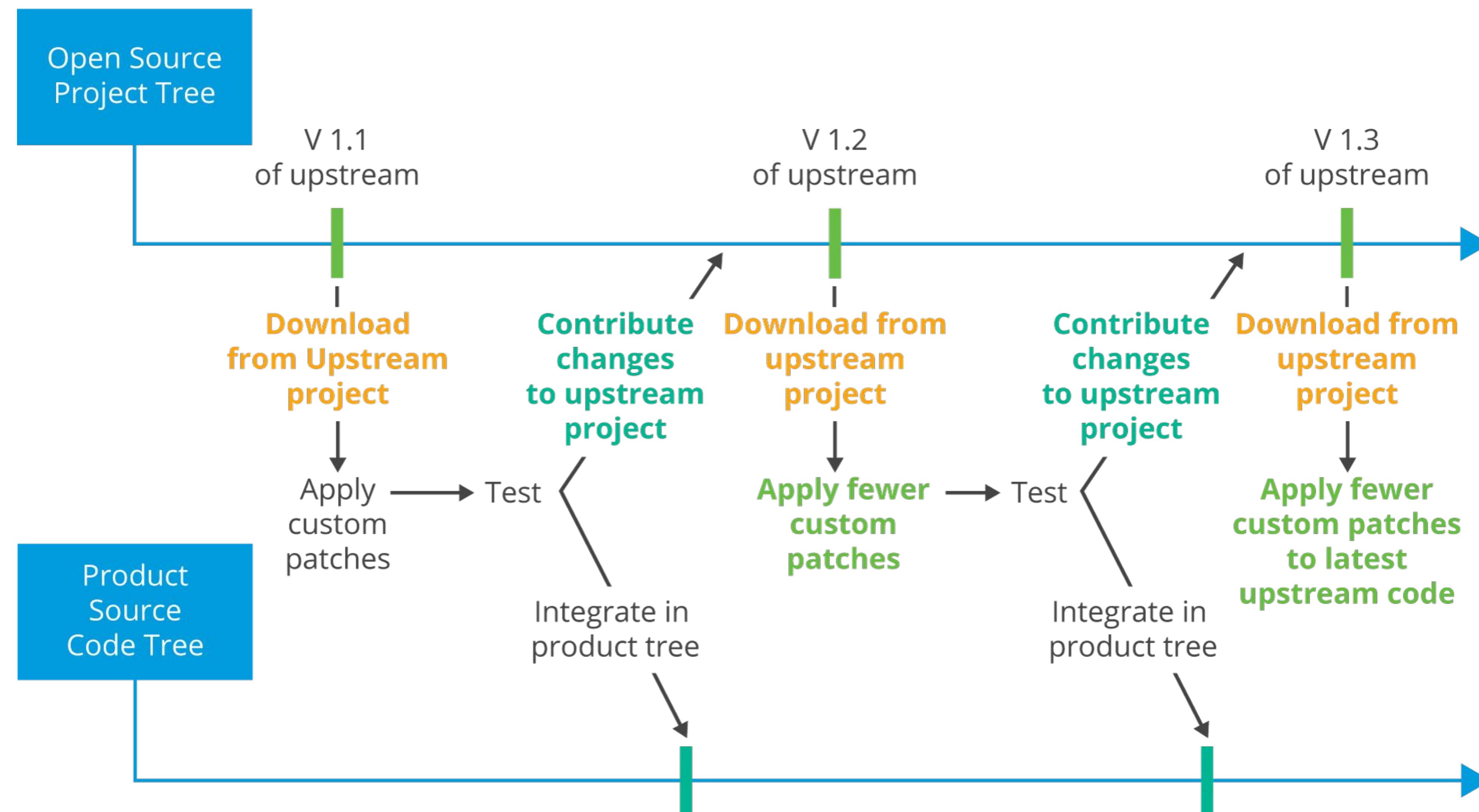


Helps organizations recruit talent from projects and retain their own developers by engaging them with the open source innovation engine

Working in a silo



Working in alignment with upstream projects, reducing technical debt



65% of organizations
that frequently
contribute code
upstream **HAVE**
A FORMALLY
STRUCTURED OSPO.



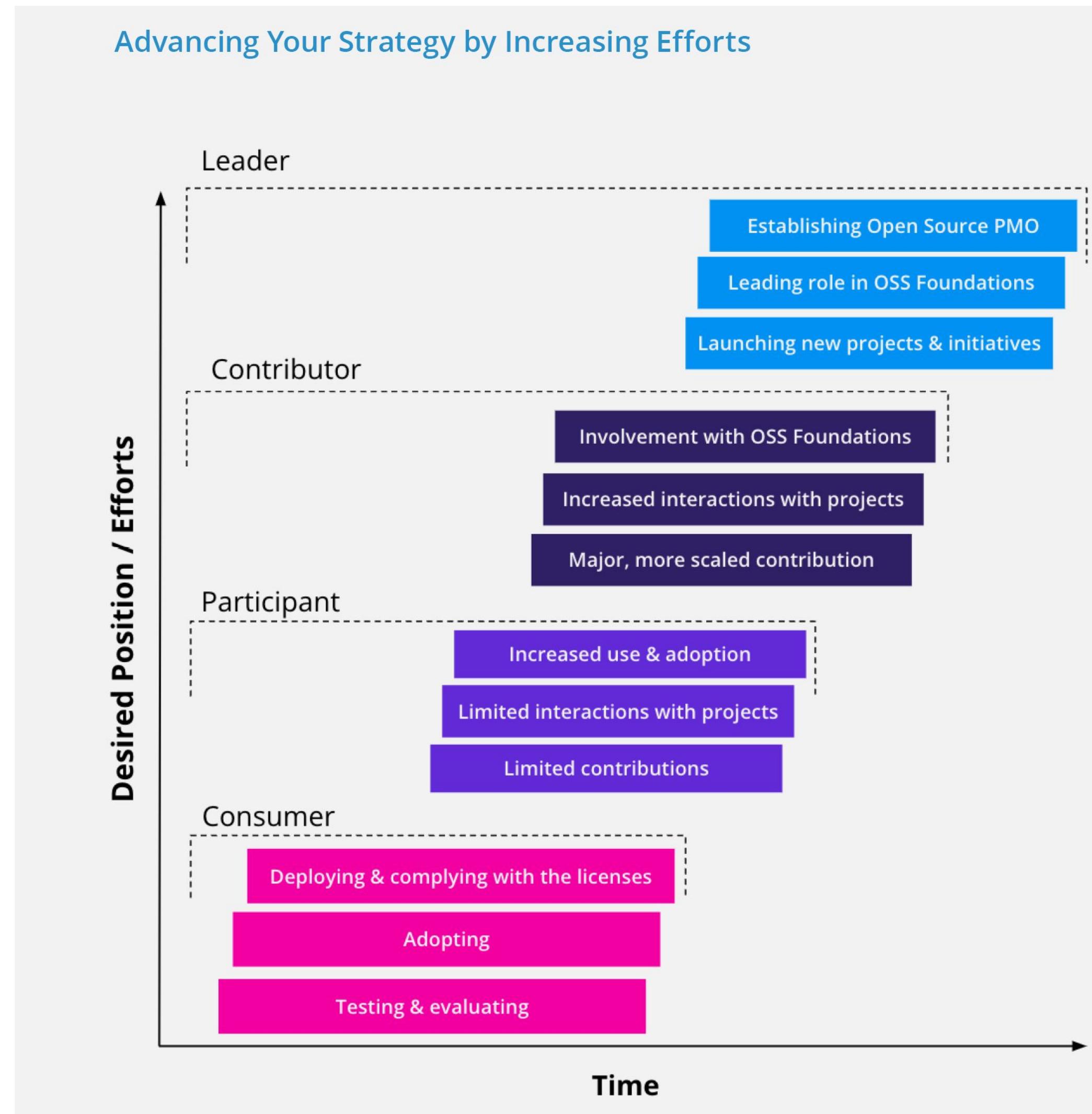
OSPO STRUCTURE

TODO Group - 2022 Survey
<https://github.com/todogroup/osposurvey/tree/main/2022>

Becoming a Leader in OSS Projects

- This scenario requires significant investment in targeted open source communities and consortia to establish leadership agenda. It will also require incremental investment primarily in engineering, product management, and legal to establish leadership in external communities and industry consortia.
- Approach for an open source leadership strategy:
 - Increase engagement with targeted open source communities
 - Selectively engage with open standards to drive the company's agenda
 - Engage with open source foundations
 - Incrementally invest in engineering, product management, and legal to engage with existing external communities

Transitioning Across the 4 Key Pillars

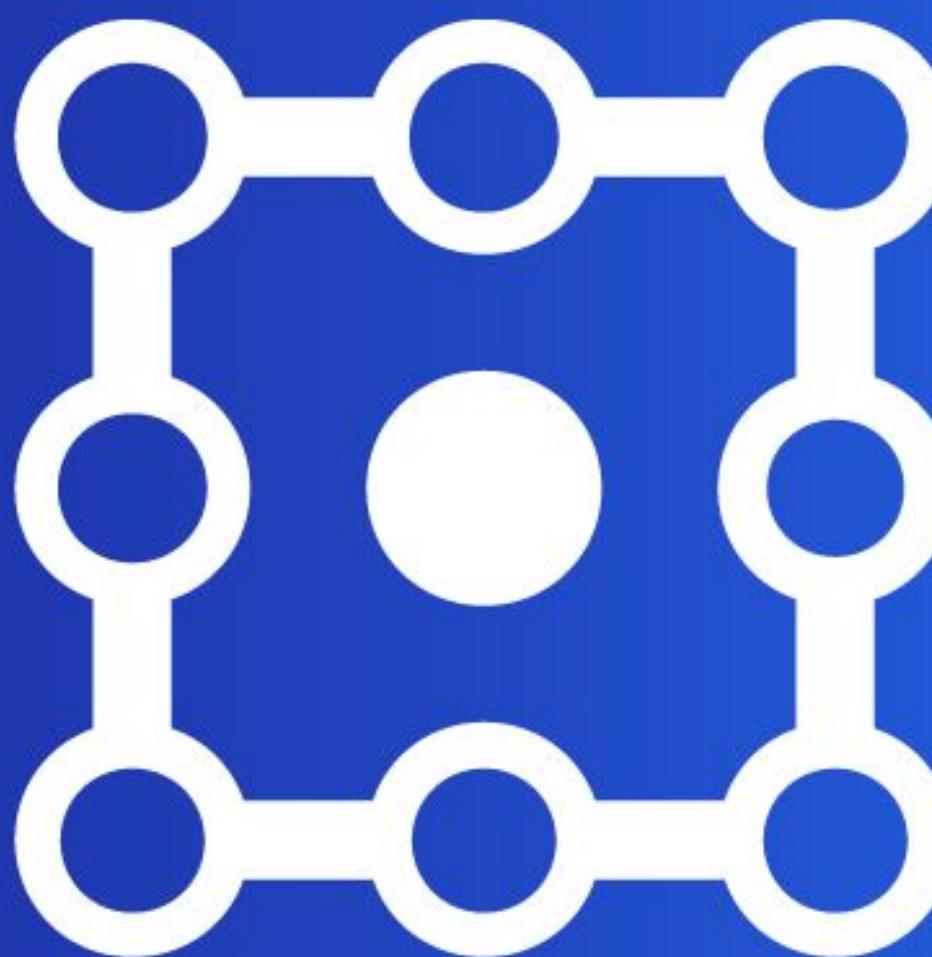


- **Leader**
 - Increase engagement with open source communities, open standards bodies, and foundations.
 - Launch new open source initiatives and increase your visibility in open source communities.
- **Contributor**
 - Hire or train developers that focus specifically on open source contributions and deploy the necessary tools to support internal open source engineering.
- **Participant**
 - Begin engaging with the open source community on communication platforms and at events.
 - Sponsor projects and organizations that are important to open source software you rely on for your products.
- **Consumer**
 - Establish internal infrastructure that enables proper open source practices and incorporates open source policies, processes, checklists, and training.

OSPO Organizational Models

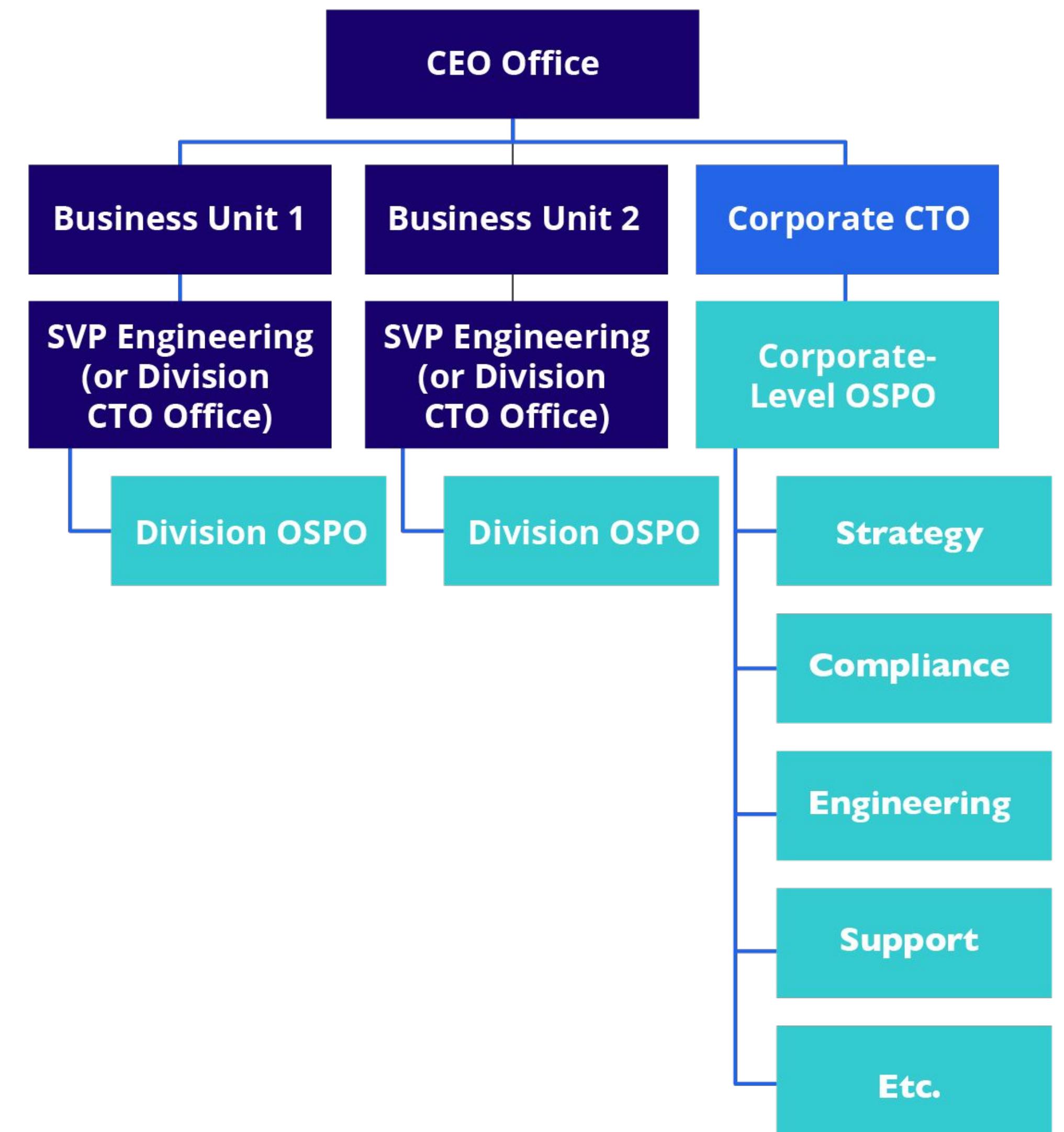
WHAT IS AN OSPO?

OSPO is designed to be **the center of competency for an organization's open source operations and structure.**

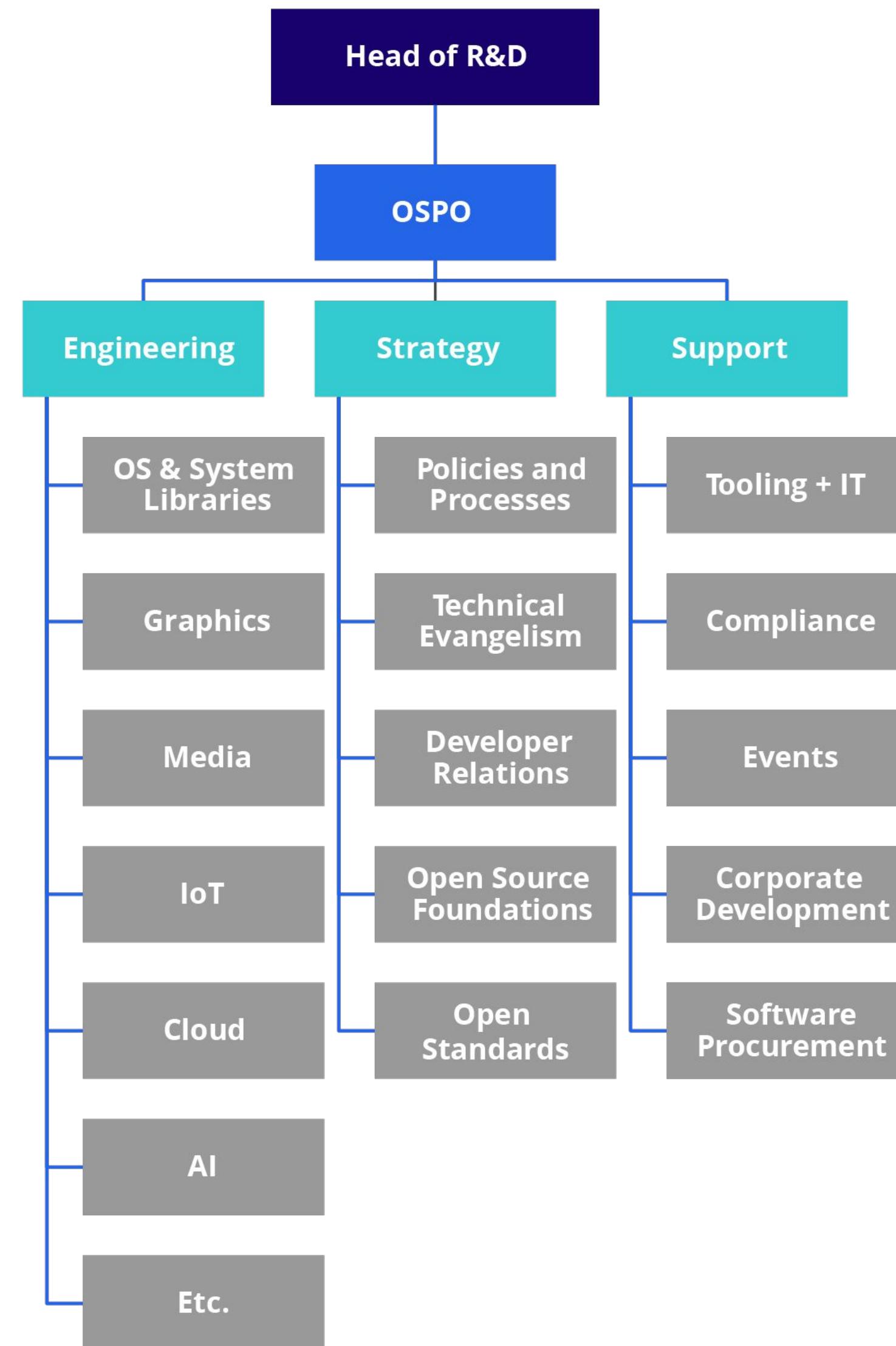


No OSPO?

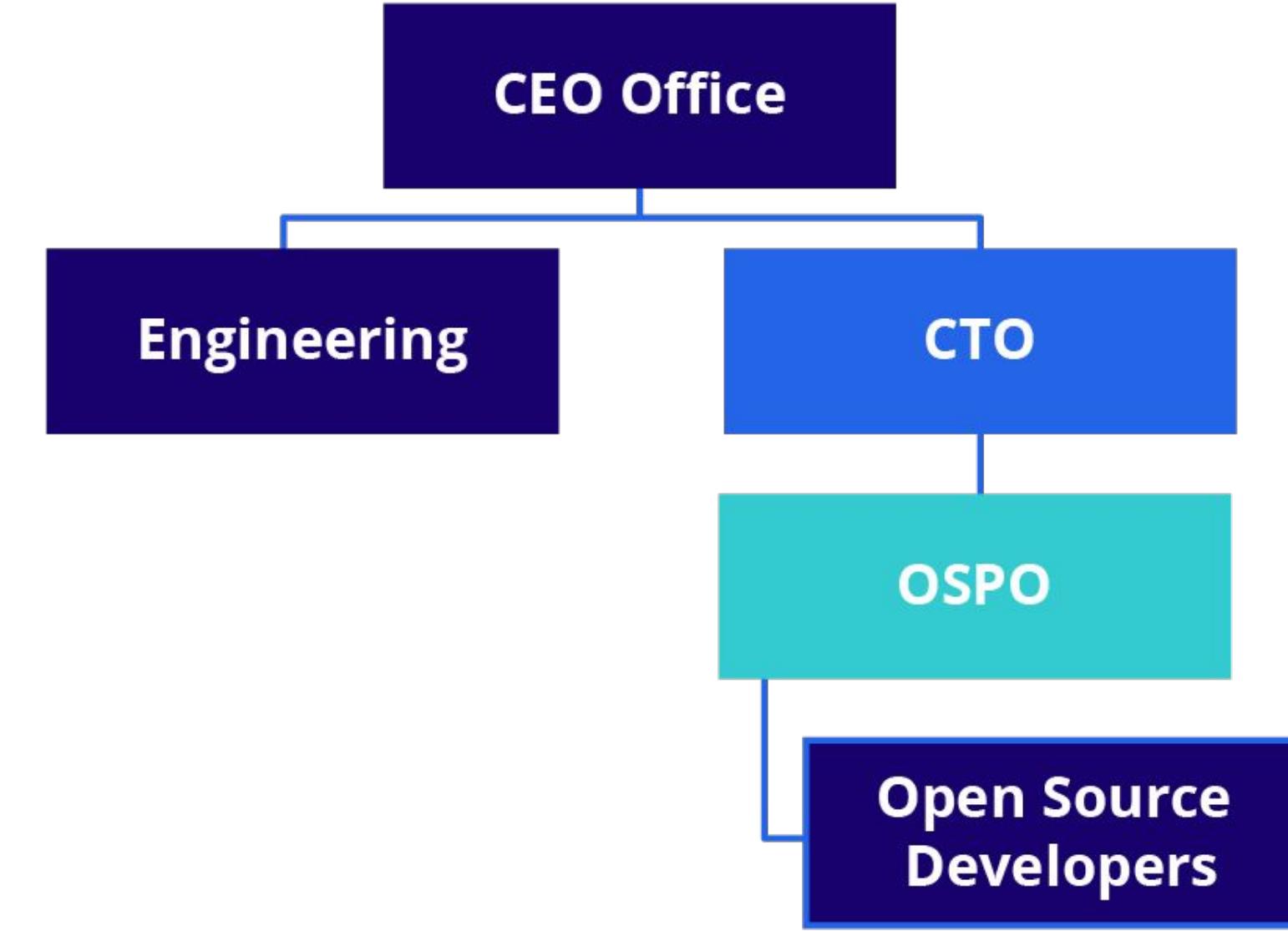
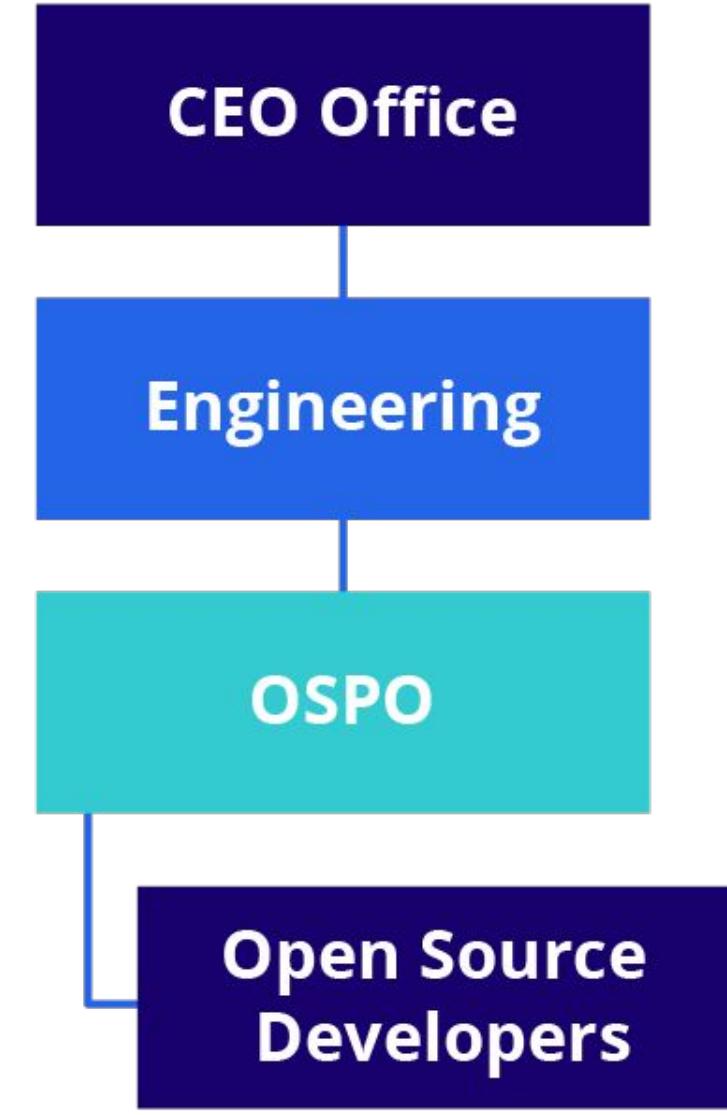
Time to establish one!



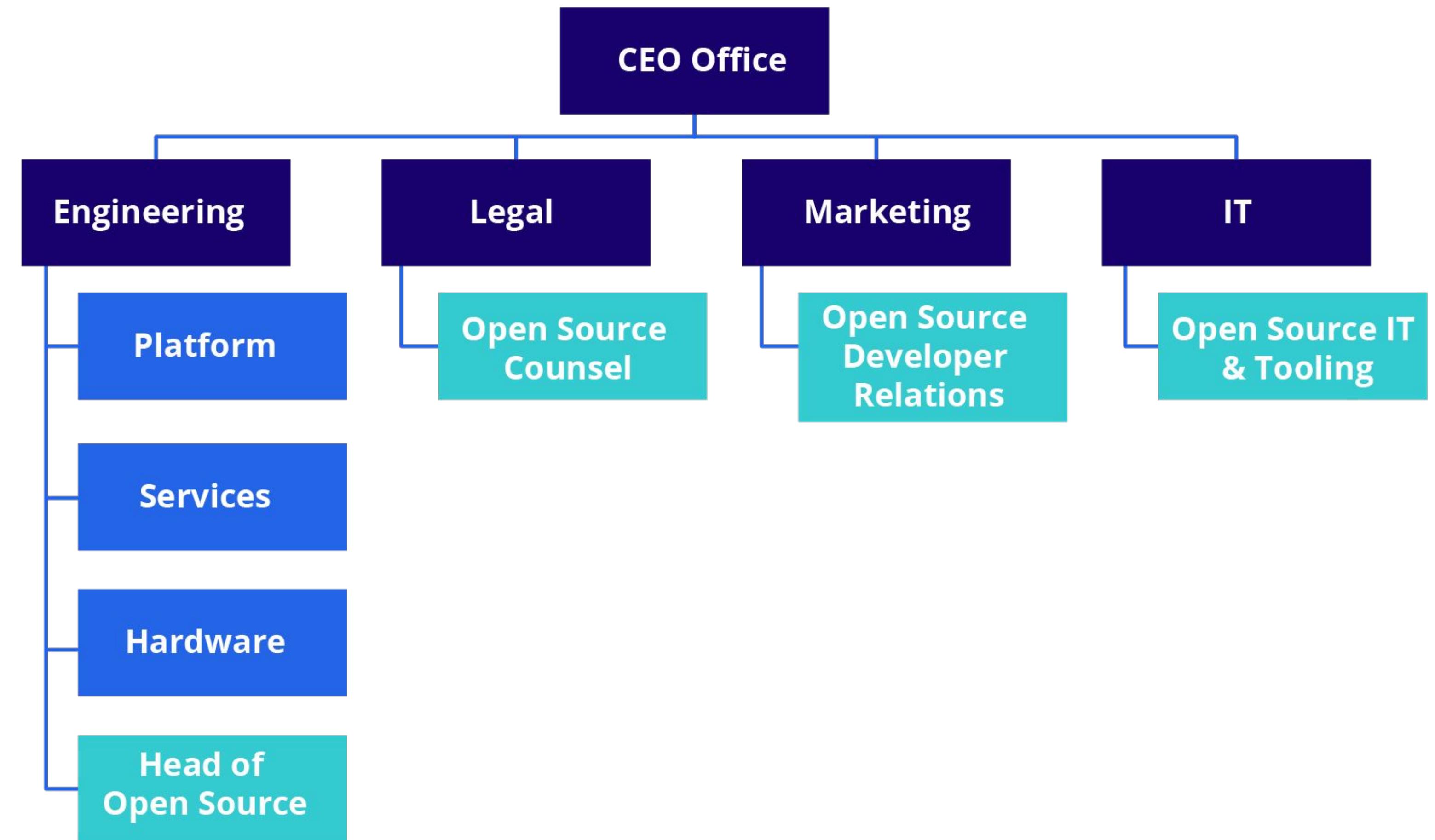
Corporate and Division-level OSPOs



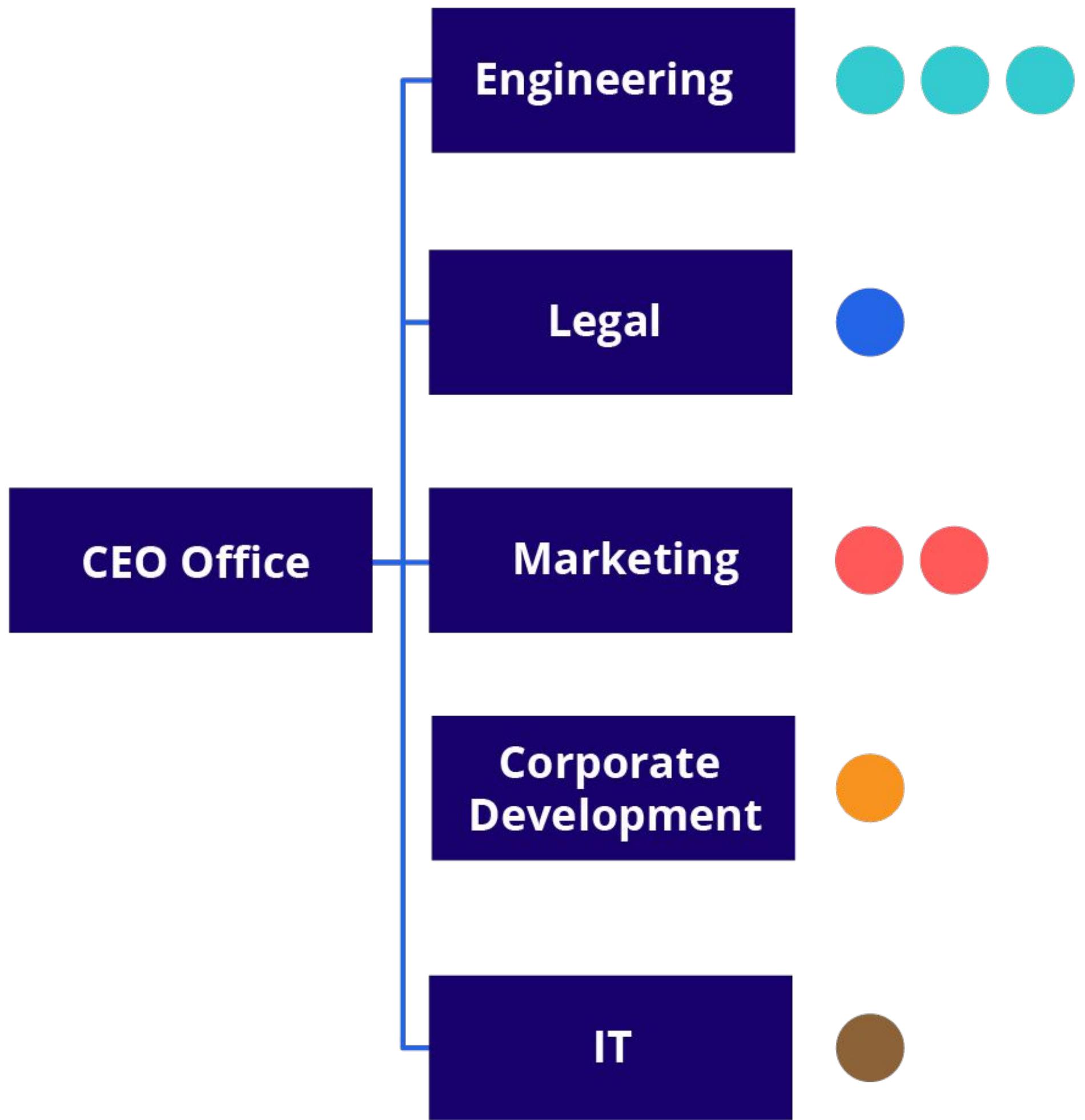
OSPO under R&D organization



OSPO part of CTO Office or under Engineering

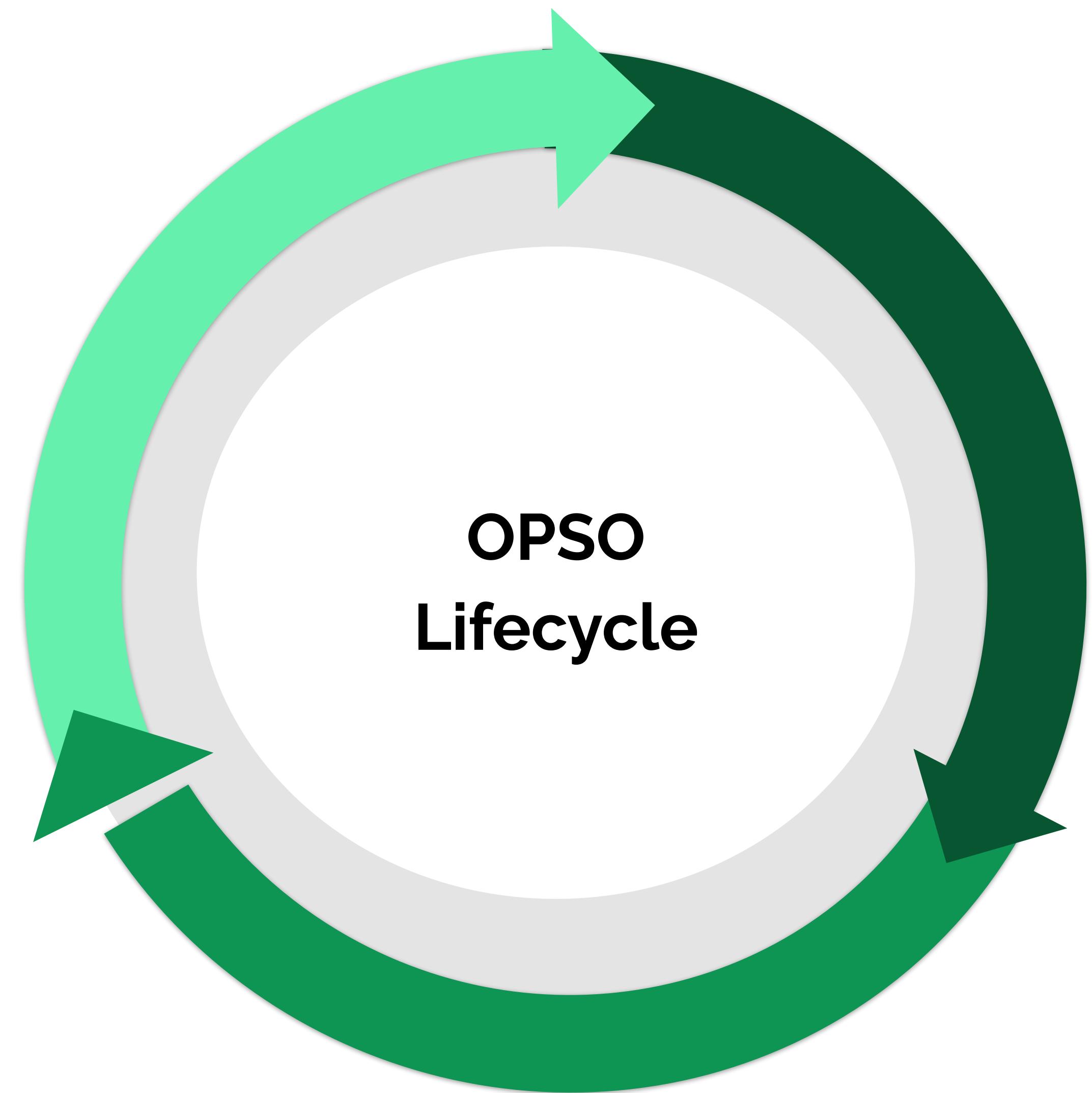


Distributed OSPO



Virtual OSPO

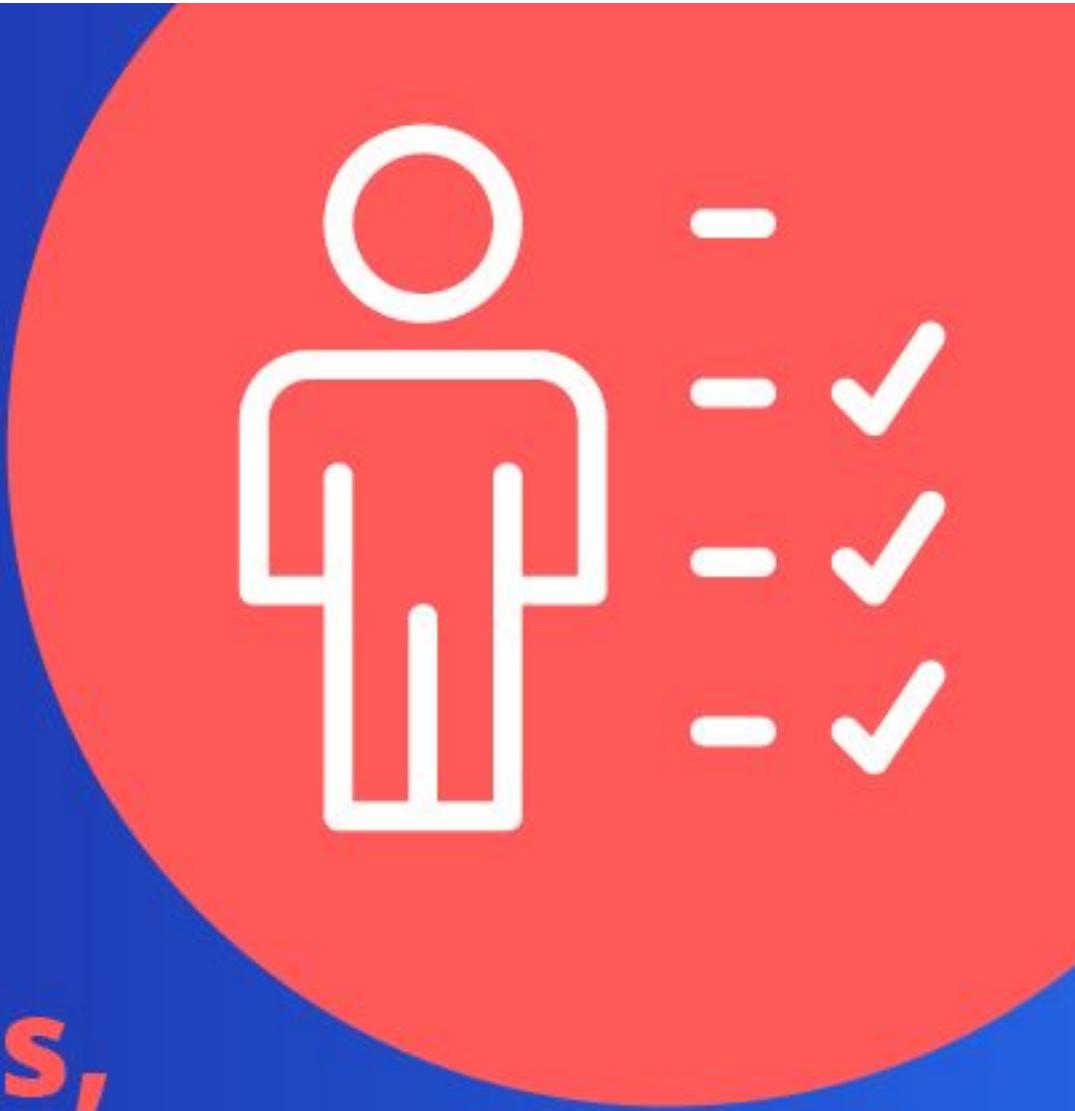
**one
size
doesn't
fit
ALL**



OSPO Staffing

OSPO STAFF

OSPOs can be run by any distinguished individuals with **strong sets of skills, such as software architects, technical evangelists, compliance engineers, and legal counsel.**



Staffing

The staffing of an OSPO depends on many variables.

- Several roles are required regardless of the specific structure of any given OSPO.
- These roles do not have to be distinct positions.

Distinguished individuals with a very strong set of skills can fulfill more than one role.

OSPO Staff

Head of OSPO

The Head of the OSPO is sometimes called Director or Vice President of Open Source, depending upon the size of the organization and the open source team. The Head of Open Source is responsible for managing and executing company-wide open source strategy and business metrics to track the business and technical success of the open source program. Depending on the structure of the OSPO, the office leader can also be responsible for open source engineering resources, ensuring open source compliance, representing the company towards open source organizations, and participating in open standards efforts.

Software Architect

We believe it is mandatory for an OSPO to have a Senior Software Architect or a Principal Engineer to act as a high-level technical decision maker on topics related to open source software, from design choices to technical standards, such as platforms and coding standards.

OSPO Staff

Technical Evangelist

A technical evangelist is an individual with a strong technical background whose primary role is to evangelize the open source contributions and solutions developed by the open source group to the company's customers, prospects, and partners, and the open source community in general. They are responsible for building demonstrations at events, delivering technical presentations, creating documentation, and generally building support to a critical mass for a given technology.

Compliance Engineer

The compliance engineer supports the execution of the company's compliance policy and process, ensuring the company fulfills all license obligations for the open source software used in products and services. Some OSPOs have complete ownership of the open source compliance function; in these cases, the OSPO may need to host multiple compliance engineers.

Legal Counsel

It is rare for an OSPO to have a Legal Counsel among its staff. In most cases, having access to a legal counsel versed in open source licensing is sufficient for small and medium size companies.

OSPO Roles & Responsibilities

CHARACTERISTICS OF OSPO MATURITY

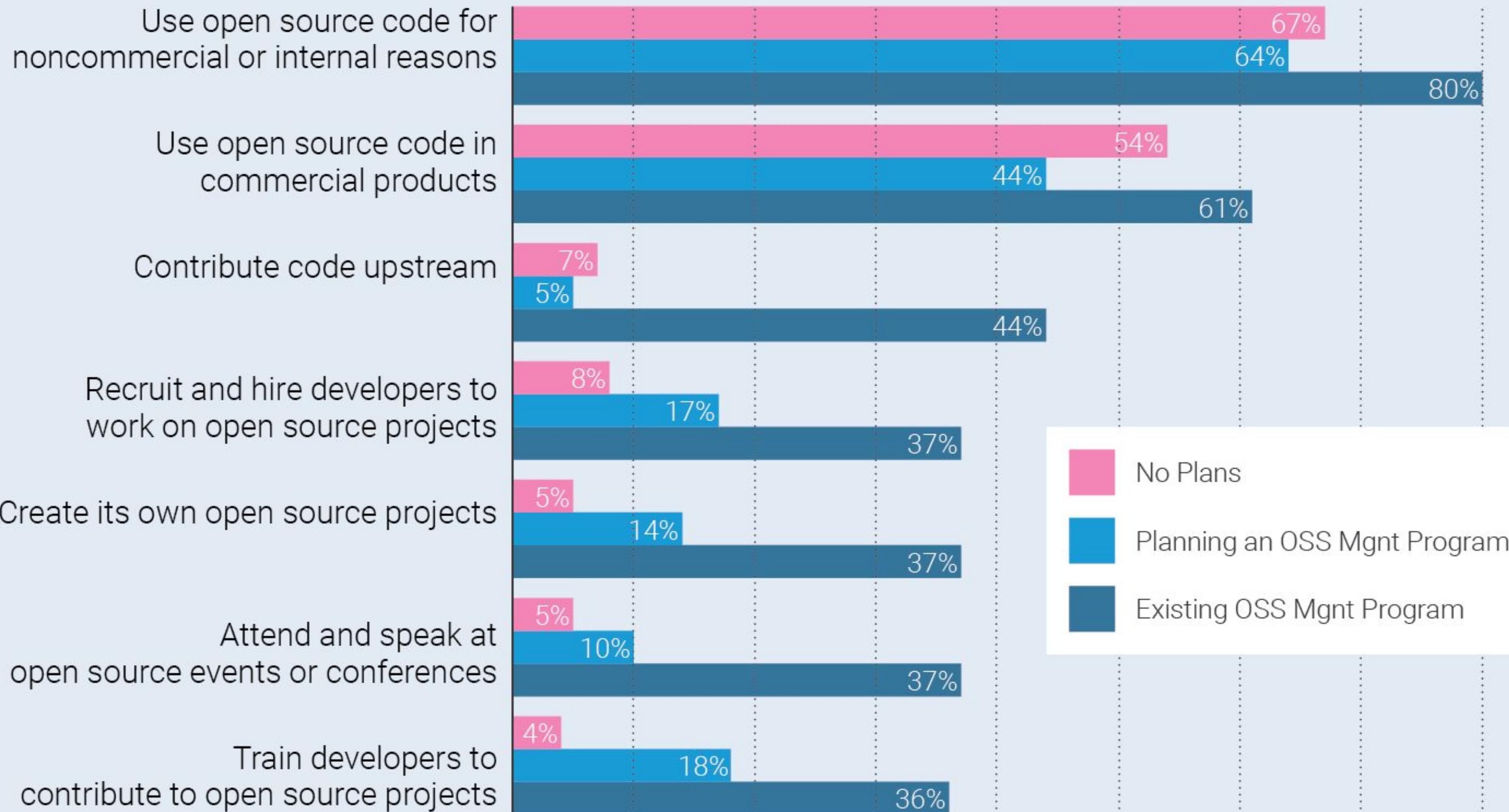
Mature OSPOs **oversee OSS consumption, governance, strategy, contribution, processes, and tooling.**



Primary responsibilities

- Develop, own and execute OSS strategy
- Oversee and maintain OSS compliance practices and policies
- Establish policies, processes, guidelines
- Prioritize and drive upstream development
- Work with open source foundations
- Track performance metrics
- Implement innersourcing practices
- Manage open source IT infrastructure
- Facilitate effective use of open source
- Grow open source talent
- Be the advisor on all OSS matters
- Eliminate friction from using or contributing to OSS
- Support corporate development activities (M&A)
- Collaborate with universities
- Foster open source culture
- Develop and deliver OSS training

Companies with OS Programs More Likely to Contribute Code, Recruit OS Developers and Create New Projects



Source: Open Source Program Survey. Q. How often does your company do the following activities?
Existing OSS Management Program, n=278; Planning an Open Source Program Management, n=119; No Plans, n=347.

Challenges

ENTERPRISE CHALLENGES

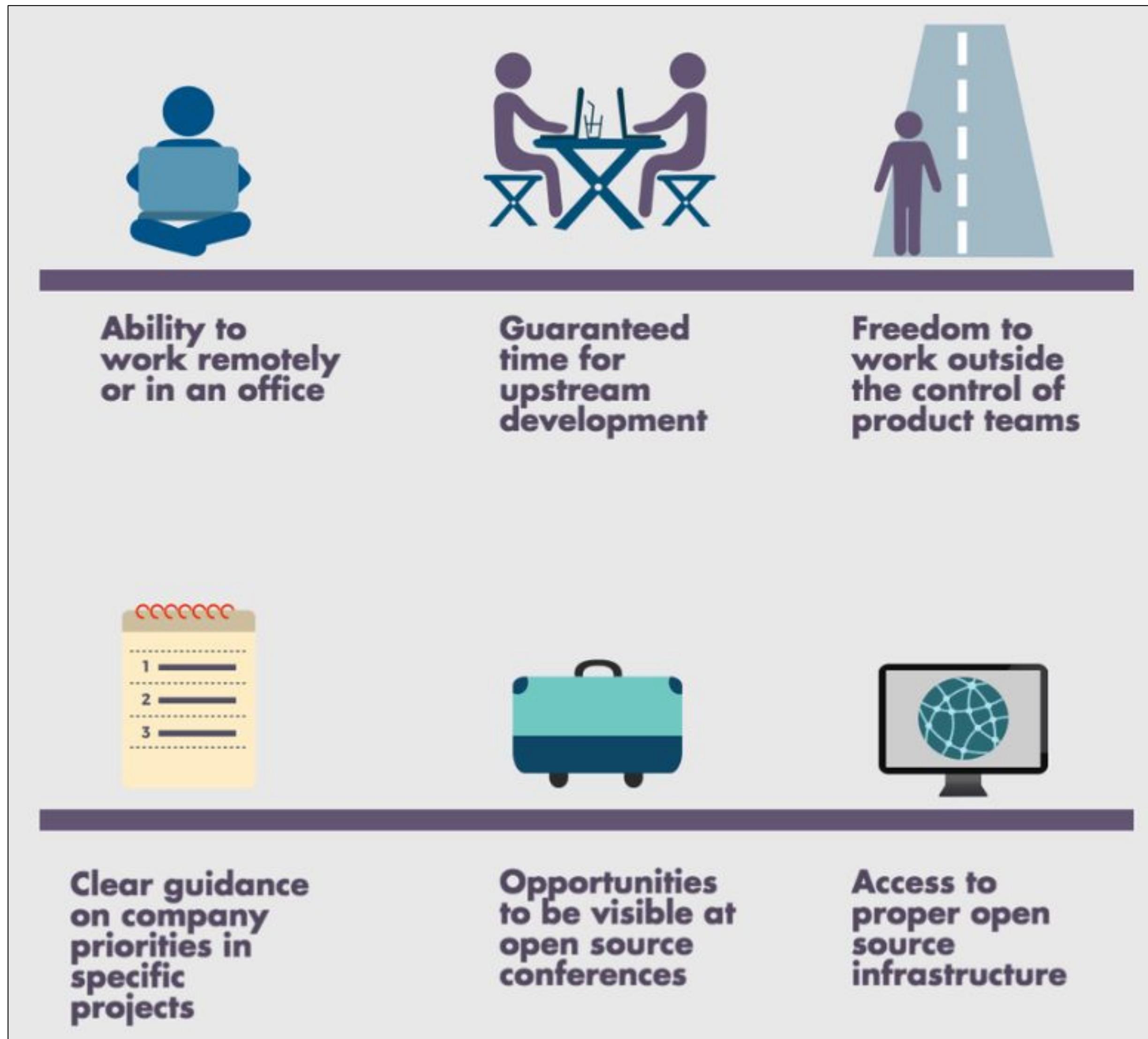
Organizations face significant challenges in open source in five specific areas: **culture, processes, tools, continuity, and education.**



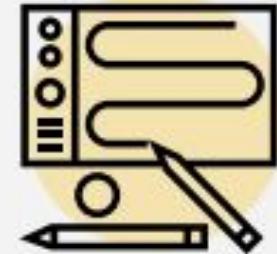
CHALLENGES ENTERPRISES FACE AS PART OF INSTITUTIONALIZING OPEN SOURCE DEVELOPMENT PRACTICES

Culture	Operations	Tools	Continuity	Education
Development model Collaboration Transparency Meritocracy Team formation Hiring practices Success metrics	Governance Usage Compliance Contribution Approvals Policies Processes	IT infrastructure Development tools Metric tracking Knowledge sharing Code reuse Software composition analysis tool adoption	Strategy Projects Priorities Funding Executive support	Executive education Knowledge transfer Technical training Compliance training Mentorship program

Factors influencing the success of OSPOs



RECOMMENDED PRACTICES FOR CONTRIBUTING TO OPEN SOURCE PROJECTS



Design & implement with upstreaming in mind to increase the likelihood of patch acceptance



Ensure the contribution improves or introduces functionality that is useful for a broad base of users



Stay involved in upstream development post merging with the upstream project



Document the code to make it easier to understand and to lower the barrier for new contributors



Upstream for the right reasons.
Upstreaming is not a code retirement strategy



Listen to feedback, and act upon it – rework the code based on the peer review process



Follow proper coding style and secure code guidelines



Follow the processes set by the project for submitting code, new features, etc.

Recommendations for making an impact

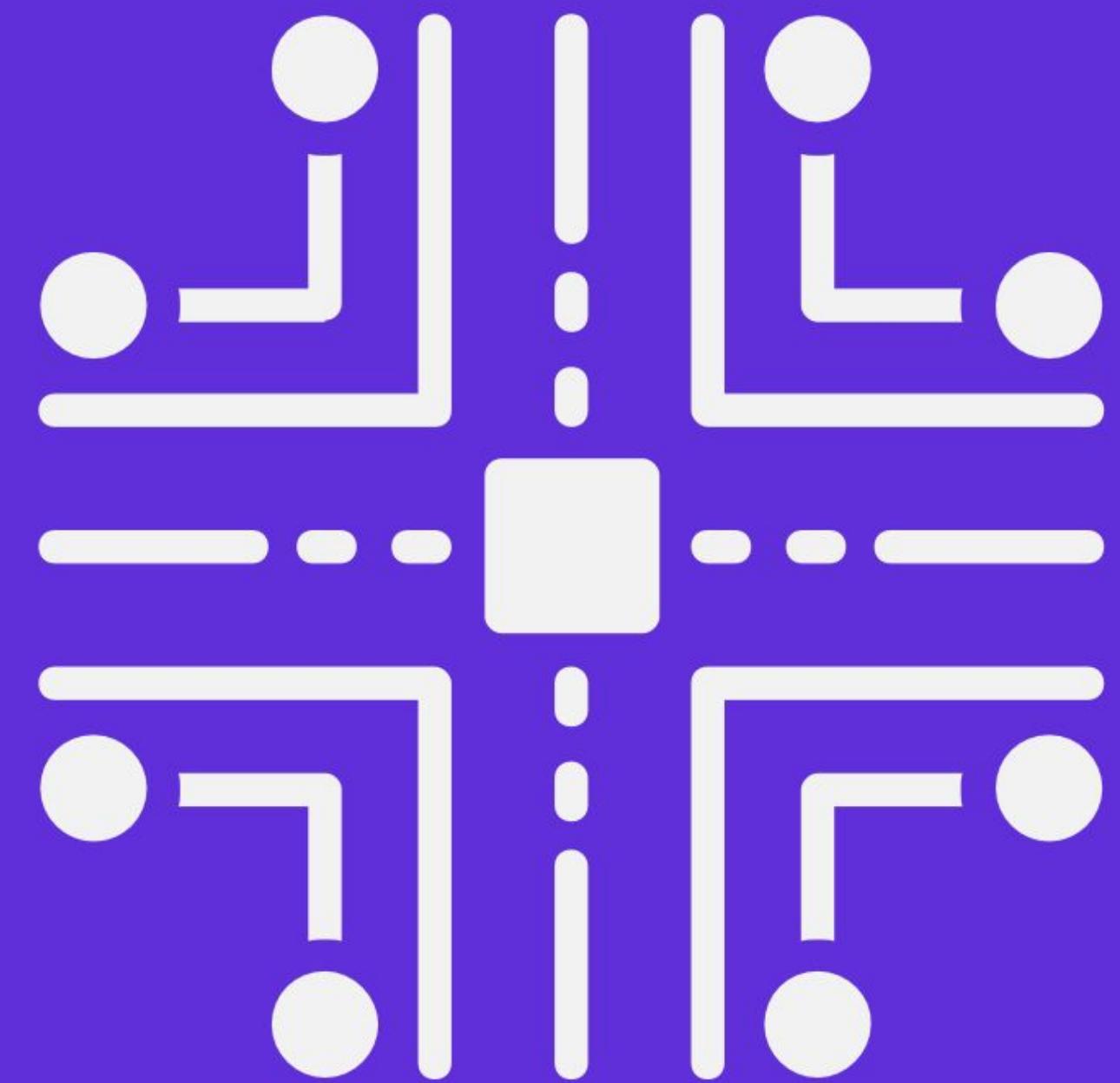
PREPARING FOR OPEN SOURCE



**IDENTIFY RELIANCE ON
OPEN SOURCE SOFTWARE,
clarify needed open source
skills, and join organizations
such as the TODO Group.**

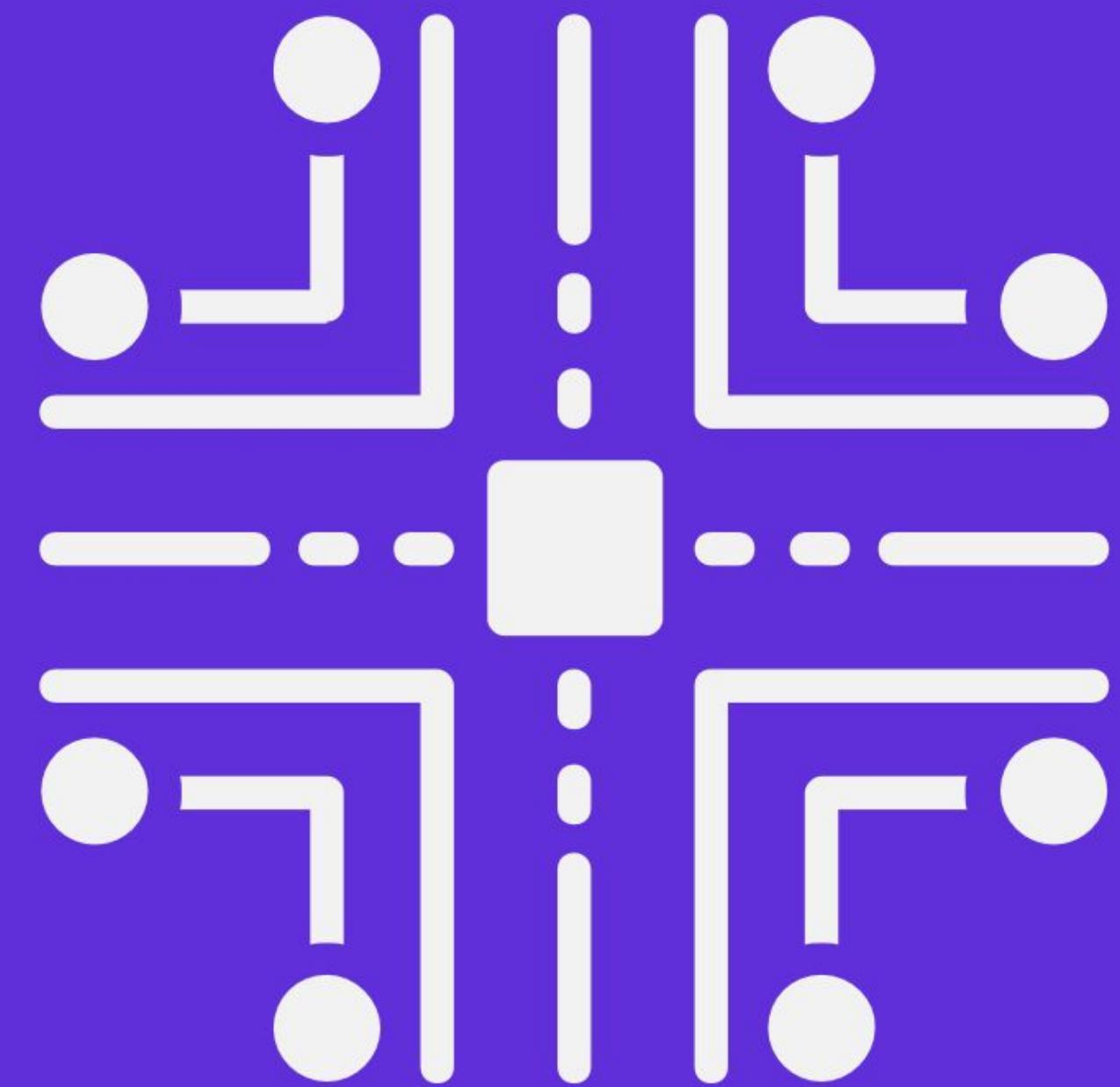
ENTERPRISE INFRASTRUCTURE

Set up your organization
to implement open
source, from compliance
to consumption to
contribution.



ENTERPRISE INFRASTRUCTURE

Set up your organization
to implement open
source, from compliance
to consumption to
contribution.



Be patient and seek out influential peers when **growing your domain expertise, open source methodology, and working practices.**





Practice and encourage **an open and collaborative mindset** when implementing open source infrastructure.

Adopt IT infrastructure
that is **flexible**
and supportive
of open source
development.





Track success through metrics that are **designed specifically for an open source environment**.



Follow a
**lightweight and
tailored approach**
to source code
contribution
approvals.

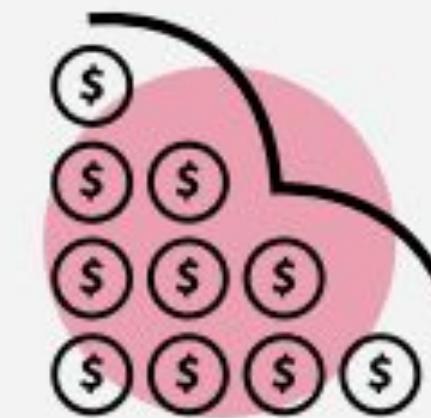


Share information across divisions and foster internal collaborations for successful implementation of innersource practices.

BENEFITS OF ADOPTING INNERSOURCE PRACTICES IN THE ENTERPRISE



Faster release cadency



Reduce costs of development



Improve source code quality



Increase internal collaboration



Increase motivation



Increase morale, retention



Increase internal information sharing

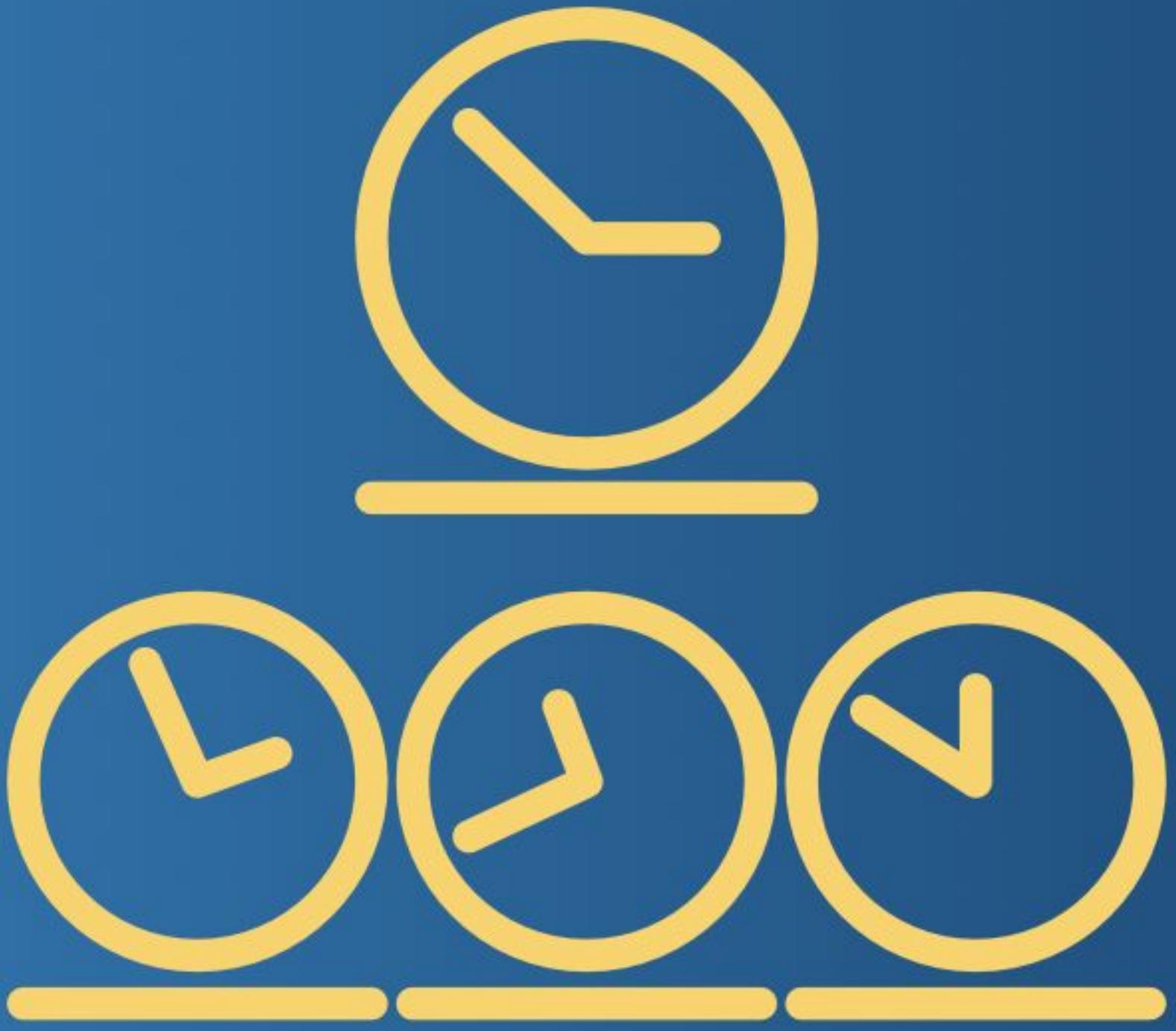


Increase internal communication

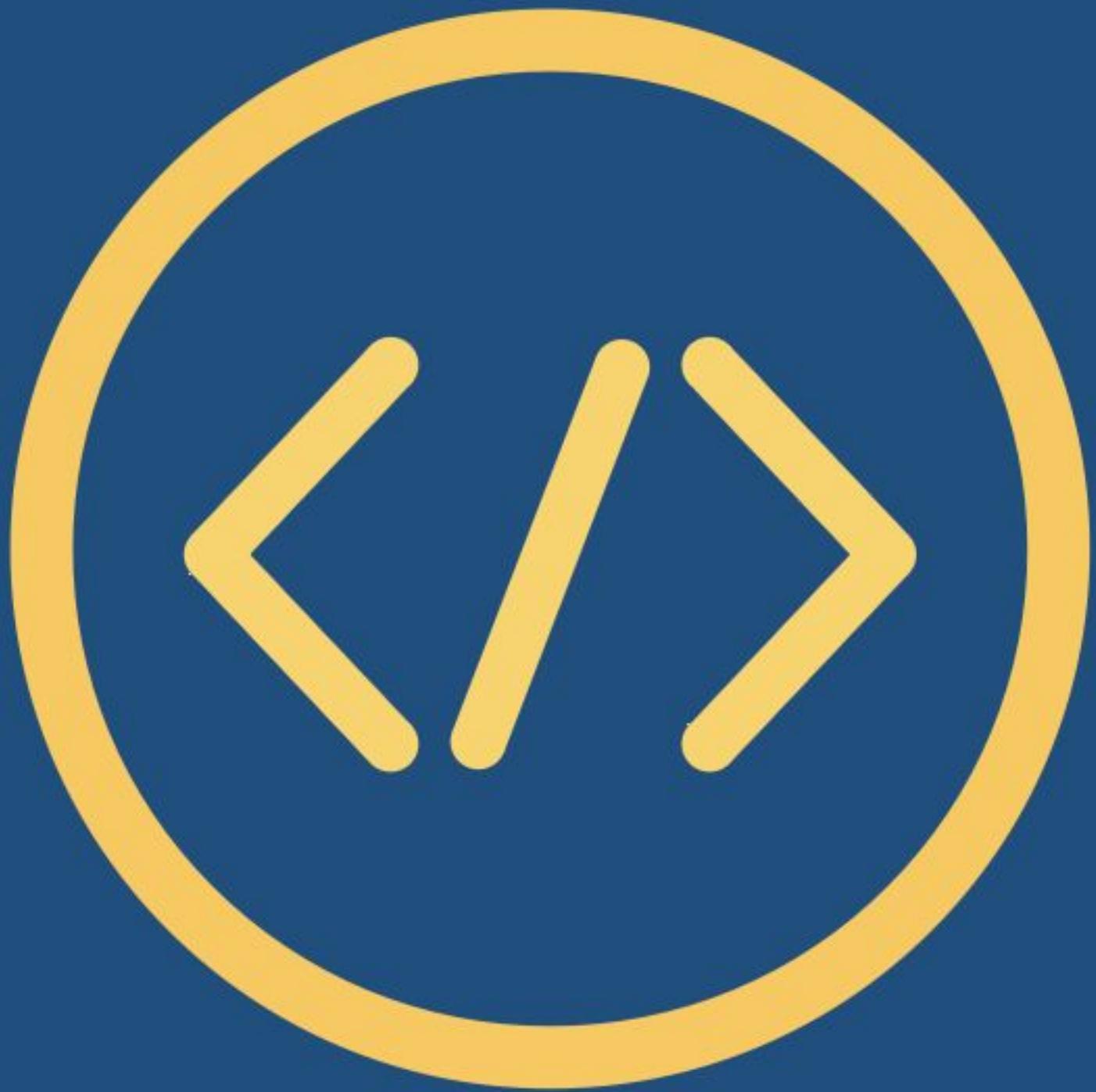
Contribute strategically to projects that are commonly used across products and services to remain essential, justifiable, and fundable.



Allocate time for
open source
developers to meet
upstream
responsibilities,
especially if they
are maintainers.



Partner with product teams on **upstream code development** that helps reduce their technical debt.





Develop open source talent internally and **encourage involvement** in open source from developers across the organization.

Create a mentorship program to support the growth of junior developers and increase the quality and quantity of code accepted in open source projects.





Participate in and host open source events **to build developer networks, participate in technical discussions, and increase visibility.**

Conclusion

MASTERING OPEN SOURCE SOFTWARE

Mastering open source software requires you to mastering the 3 critical Cs:

Consumption

Establish internal infrastructure to enable proper practices for open source software consumption: policy, process, checklists, and training.

Compliance

Enable open source compliance practices within your development process to ensure proper fulfillment of open source license obligations one products ship.

Contribution

Enable your developers to engage within open source projects via a policy and a lightweight process, and access to Legal support. Provide training on open source development model and best practices.

Principles to Embrace

1

We can't hire all the smart people in the world.

We need to find ways to collaborate with them.

2

Open source R&D creates significant value.

Internal R&D claims portion of that value.

3

We don't need to create the project to benefit from it.

We can join existing project and excel in it.

How can you build a healthy environment for open source consumption?

- 1 Establish a policy and process to guide open source usage
- 2 Set up a team to oversee approvals for all open source usage
- 3 Understand your open source product strategy and core values
- 4 Provide the enabling IT infrastructure and tooling
- 5 Setup an open source license compliance program
- 6 Offer training to your staff and manager
- 7 Track everything, measure, improve and communicate
- 8 Adopt open source practices for your internal development
- 9 Identify incoming open source code through your software suppliers
- 10 Identify key open source projects and start contributing to them

How can you build a healthy environment for open source contributions?

- 1 Establish a policy and process to guide open source contributions
- 2 Set up a team to oversee approvals for all open source contributions
- 3 Focus contributions in the areas that will enable your technologies
- 4 Provide the needed IT infrastructure and tooling for contributors
- 5 Offer training to your staff on contribution best practices
- 6 Track contributions, measure impact, improve and communicate
- 7 Establish a mentorship program to train less experienced developers
- 8 Provide contributions guidelines, How-To's, Do's and Don'ts
- 9 Make open source legal support accessible to developers
- 10 Hire from the open source communities you value the most
- 11 Always follow community processes/practices of specific projects

Resources



Releasing Internal Code into a New Open Source Project

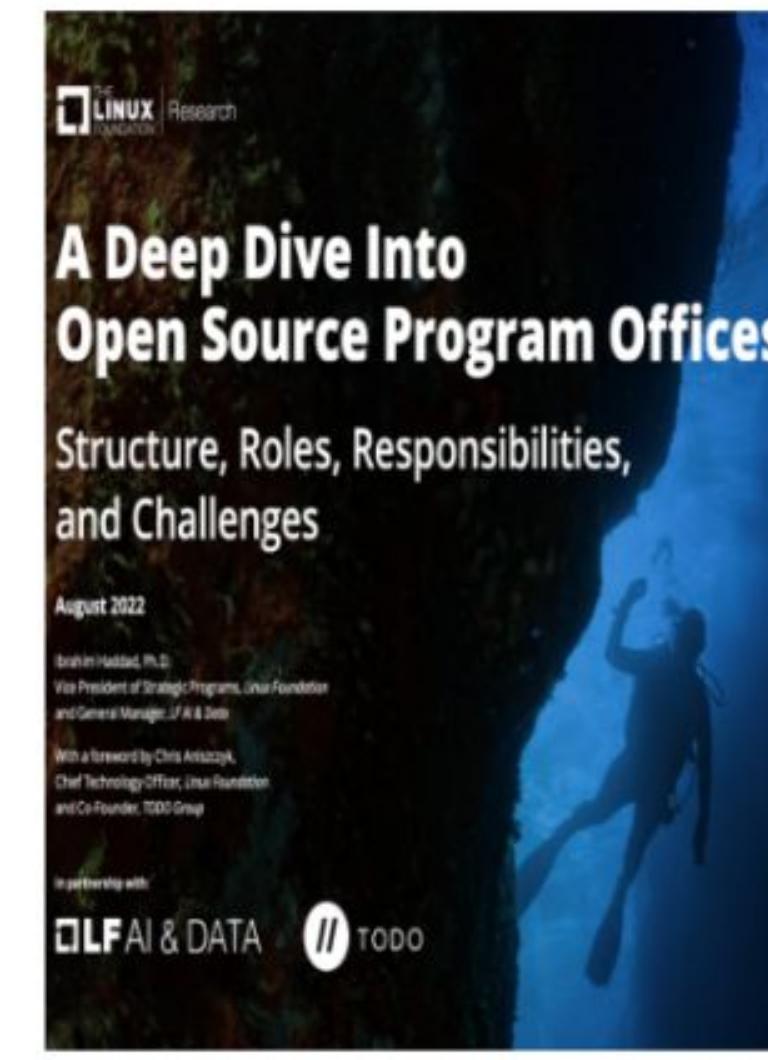
November, 2022



A Road Map to
Improve the
Effectiveness and Impact of Enterprise
Open Source
Development

February 2023

Ibrahim Haddad, Ph.D.
Executive Director, LF AI & Data and PyTorch Foundation
Foreword by Jessica Murillo
VP and Delivery Practice Leader, IBM



A Deep Dive into Open Source Program Offices

August, 2022

February 2023



Guide to Enterprise Open Source (Chinese Edition)

July, 2022



Guide to Enterprise Open Source

May, 2022



Artificial Intelligence and Data in Open Source

March, 2022



Technical Debt and Open Source Development

July, 2020



A Roadmap to Improve the Effectiveness and Impact of Open Source Development

Download PDFs:
<https://www.ibrahimlinux.com/books/>

Building and Managing an Open Source Program Office

Ibrahim Haddad, Ph.D.
Vice President, Strategic Programs
Linux Foundation



@IbrahimAtLinux



IbrahimAtLinux.com



linkedin.com/in/ibrahimhaddad/