

Design a DevOps Strategy for Multi-Org Implementations



Sebastiano Schwarz & Christopher Ramm



Christopher Ramm

DCX CTO Germany,
Capgemini



Sebastiano Schwarz

Salesforce CTO Germany,
Capgemini

Coffee on us.

The first 700 Trailblazers to provide feedback on this event will receive a \$10 Starbucks gift card.*

- 1 Download the Salesforce Events app.
- 2 Scan the QR code or navigate to My Event, then My Surveys.
- 3 Take (4) Session Surveys and the Event Survey (available on Thursday).
- 4 Redeem your gift card at Badge Pickup on Thursday.

*Restrictions apply. See terms and conditions at sforce.co/survey-terms



Forward-looking statements

The Salesforce logo, which consists of the word "salesforce" in a white, lowercase, sans-serif font, enclosed within a blue cloud-like shape.

This presentation contains forward-looking statements about, among other things, trend analyses and future events, future financial performance, anticipated growth, industry prospects, environmental, social and governance goals, and the anticipated benefits of acquired companies. The achievement or success of the matters covered by such forward-looking statements involves risks, uncertainties and assumptions. If any such risks or uncertainties materialize or if any of the assumptions prove incorrect, Salesforce's results could differ materially from the results expressed or implied by these forward-looking statements. The risks and uncertainties referred to above include those factors discussed in Salesforce reports filed from time to time with the Securities and Exchange Commission, including, but not limited to: impact of, and actions we may take in response to, the COVID-19 pandemic, related public health measures and resulting economic downturn and market volatility; our ability to maintain security levels and service performance meeting the expectations of our customers, and the resources and costs required to avoid unanticipated downtime and prevent, detect and remediate performance degradation and security breaches; the expenses associated with our data centers and third-party infrastructure providers; our ability to secure additional data center capacity; our reliance on third-party hardware, software and platform providers, the effect of evolving domestic and foreign government regulations, including those related to the provision of services on the Internet, those related to accessing the Internet, and those addressing data privacy, cross-border data transfers and import and export controls; current and potential litigation involving us or our industry, including litigation involving acquired entities such as Tableau Software, Inc. and Slack Technologies, Inc., and the resolution or settlement thereof, regulatory developments and regulatory investigations involving us or affecting our industry; our ability to successfully introduce new services and product features, including any efforts to expand our services; the success of our strategy of acquiring or making investments in complementary businesses, joint ventures, services, technologies and intellectual property rights; our ability to complete, on a timely basis or at all, announced transactions; our ability to realize the benefits from acquisitions, strategic partnerships, joint ventures and investments, including our July 2021 acquisition of Slack Technologies, Inc., and successfully integrate acquired businesses and technologies; our ability to compete in the markets in which we participate; the success of our business strategy and our plan to build our business, including our strategy to be a leading provider of enterprise cloud computing applications and platforms; our ability to execute our business plans, our ability to continue to grow unearned revenue and remaining performance obligation; the pace of change and innovation in enterprise cloud computing services; the seasonal nature of our sales cycles; our ability to limit customer attrition and costs related to those efforts; the success of our international expansion strategy, the demands on our personnel and infrastructure resulting from significant growth in our customer base and operations, including as a result of acquisitions; our ability to preserve our workplace culture, including as a result of our decisions regarding our current and future office environments or work-from-home policies; our dependency on the development and maintenance of the infrastructure of the Internet; our real estate and office facilities strategy and related costs and uncertainties, fluctuations in, and our ability to predict, our operating results and cash flows; the variability in our results arising from the accounting for term license revenue products; the performance and fair value of our investments in complementary businesses through our strategic investment portfolio; the impact of future gains or losses from our strategic investment portfolio, including gains or losses from overall market conditions that may affect the publicly traded companies within our strategic investment portfolio; our ability to protect our intellectual property rights; our ability to develop our brands; the impact of foreign currency exchange rate and interest rate fluctuations on our results; the valuation of our deferred tax assets and the release of related valuation allowances; the potential availability of additional tax assets in the future; the impact of new accounting pronouncements and tax laws; uncertainties affecting our ability to estimate our tax rate; uncertainties regarding our tax obligations in connection with potential jurisdictional transfers of intellectual property, including the tax rate, the timing of the transfer and the value of such transferred intellectual property, uncertainties regarding the effect of general economic and market conditions; the impact of geopolitical events; uncertainties regarding the impact of expensing stock options and other equity awards; the sufficiency of our capital resources, the ability to execute our Share Repurchase Program; our ability to comply with our debt covenants and lease obligations; the impact of climate change, natural disasters and actual or threatened public health emergencies; and our ability to achieve our aspirations, goals and projections related to our environmental, social and governance initiatives.

**Nobody ever simplified their
Salesforce landscape by
adding another org.**



Agenda



01 Why Multi-Org?

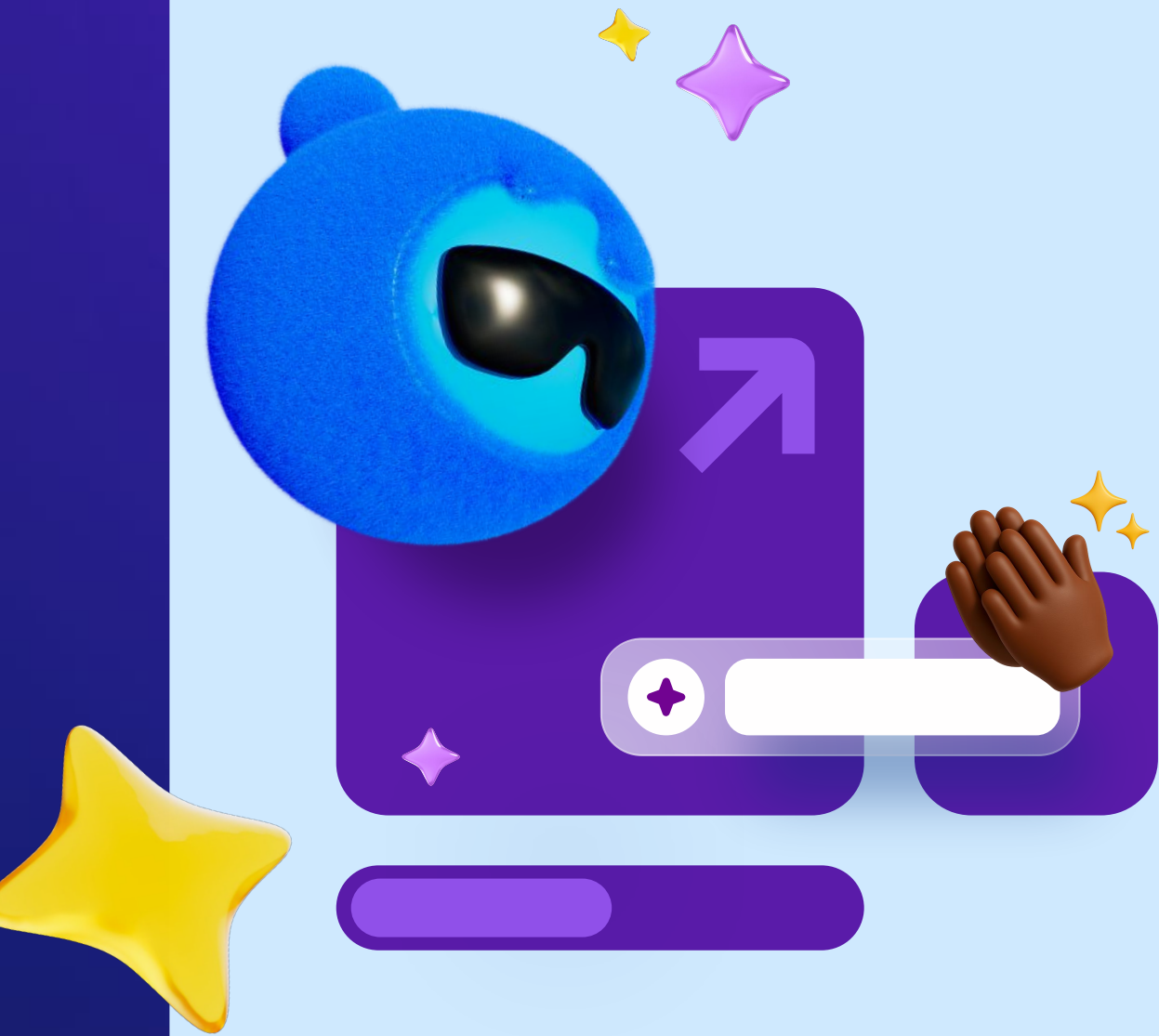
02 DevOps Approaches

03 Package-based
delivery

04 Source-driven delivery

05 Our Experience

06 Q&A



Why Multi-Org?

Five Dimensions of a Holistic Approach



Most pillars are in the organization's control

BUSINESS SETUP

- ✓ Strategy
- ✓ Business Unit setup
- ✓ Operation Model

GOVERNANCE AND CULTURE

- ✓ Sponsorship
- ✓ Business and IT Governance
- ✓ Vigilant

DATA STRATEGY

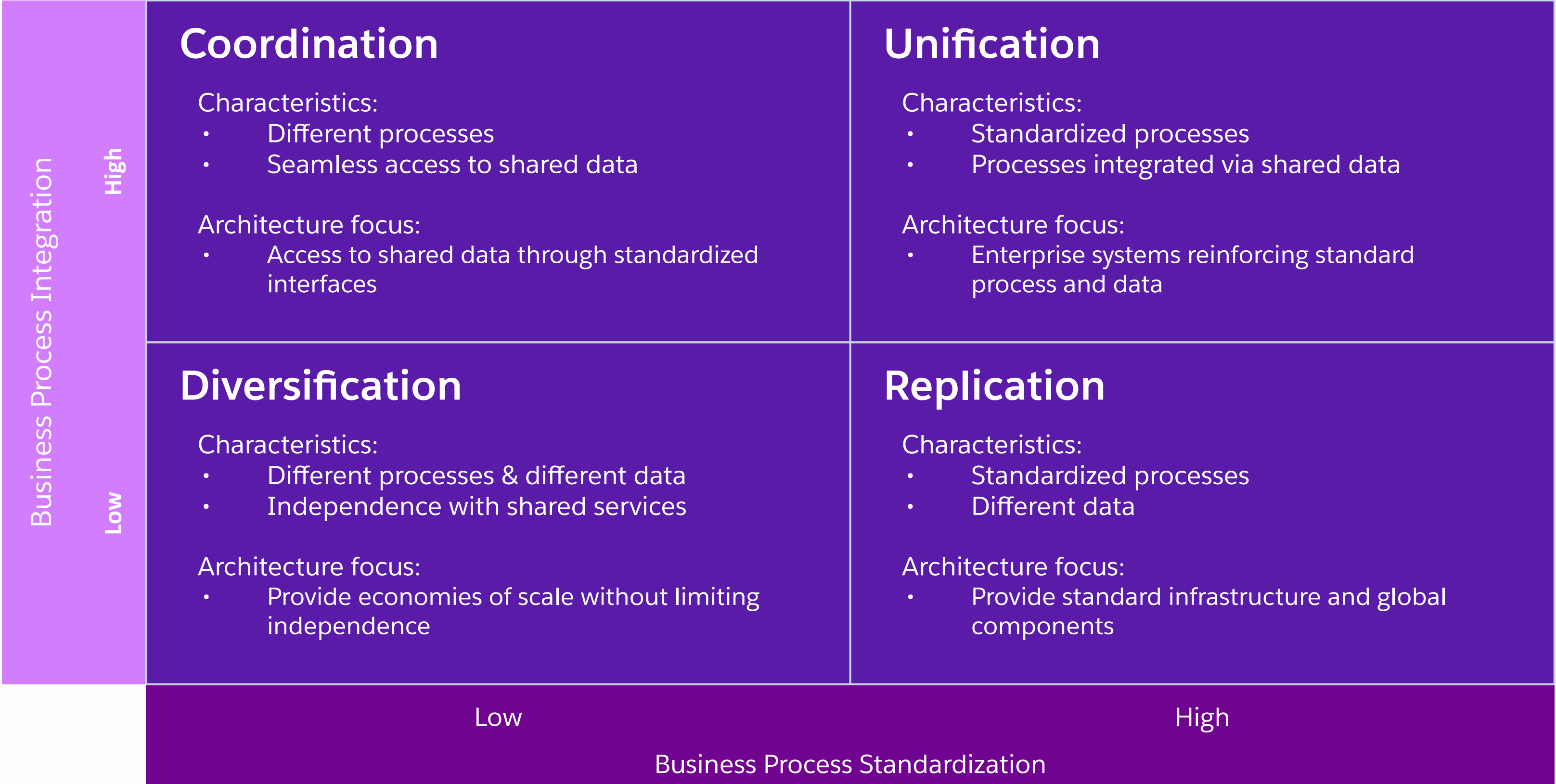
- ✓ Application architecture
- ✓ Functionality
- ✓ Data strategies
- ✓ Salesforce limitations

OPERATIONS (BUSINESS & IT)

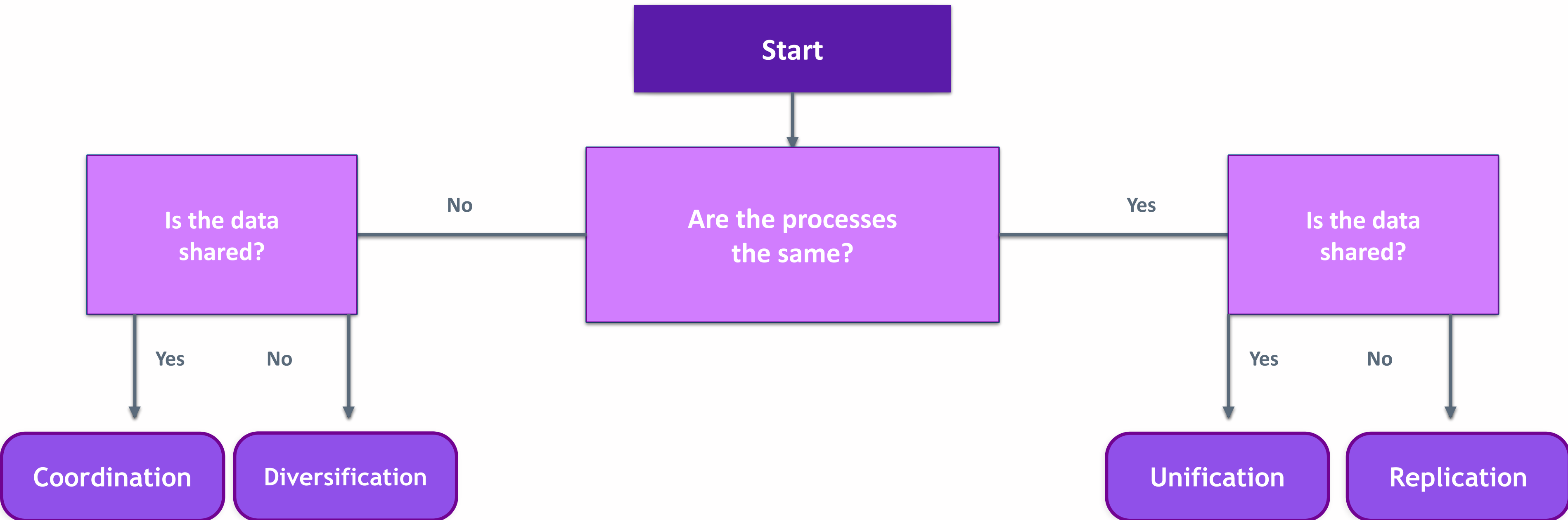
- ✓ Design & delivery approach
- ✓ Deployment + end-user support
- ✓ DevOps

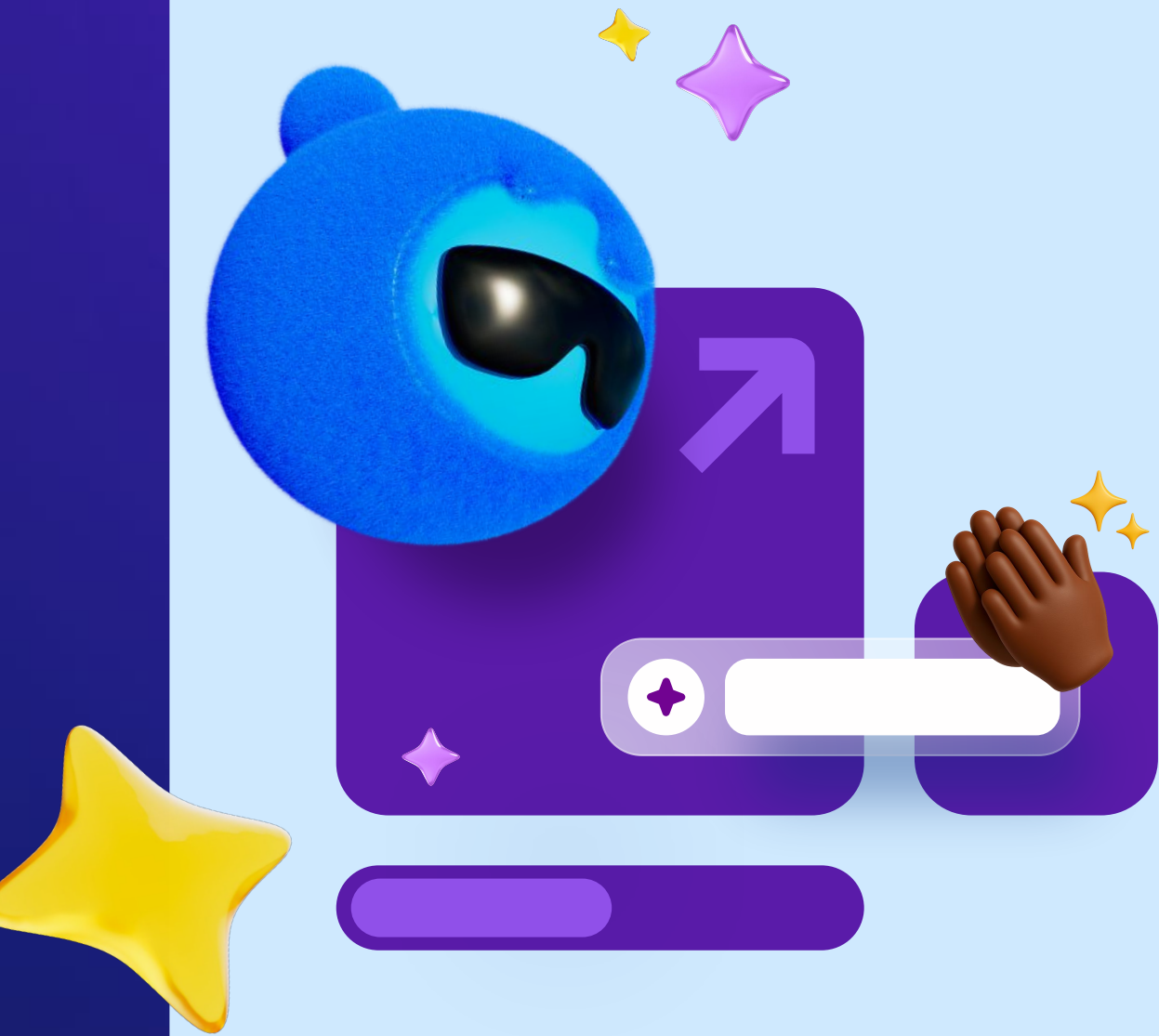
LEGAL REQUIREMENTS (MOSTLY OUT OF CONTROL. E.G: CHINA, GDPR,...)

SOGAF Operating Models



Operating Model Decision Tree





DevOps Approaches

Common Challenges in Salesforce DevOps

Cross-functional team setups

Modern DevOps practices require technical maturity that not all team members possess.

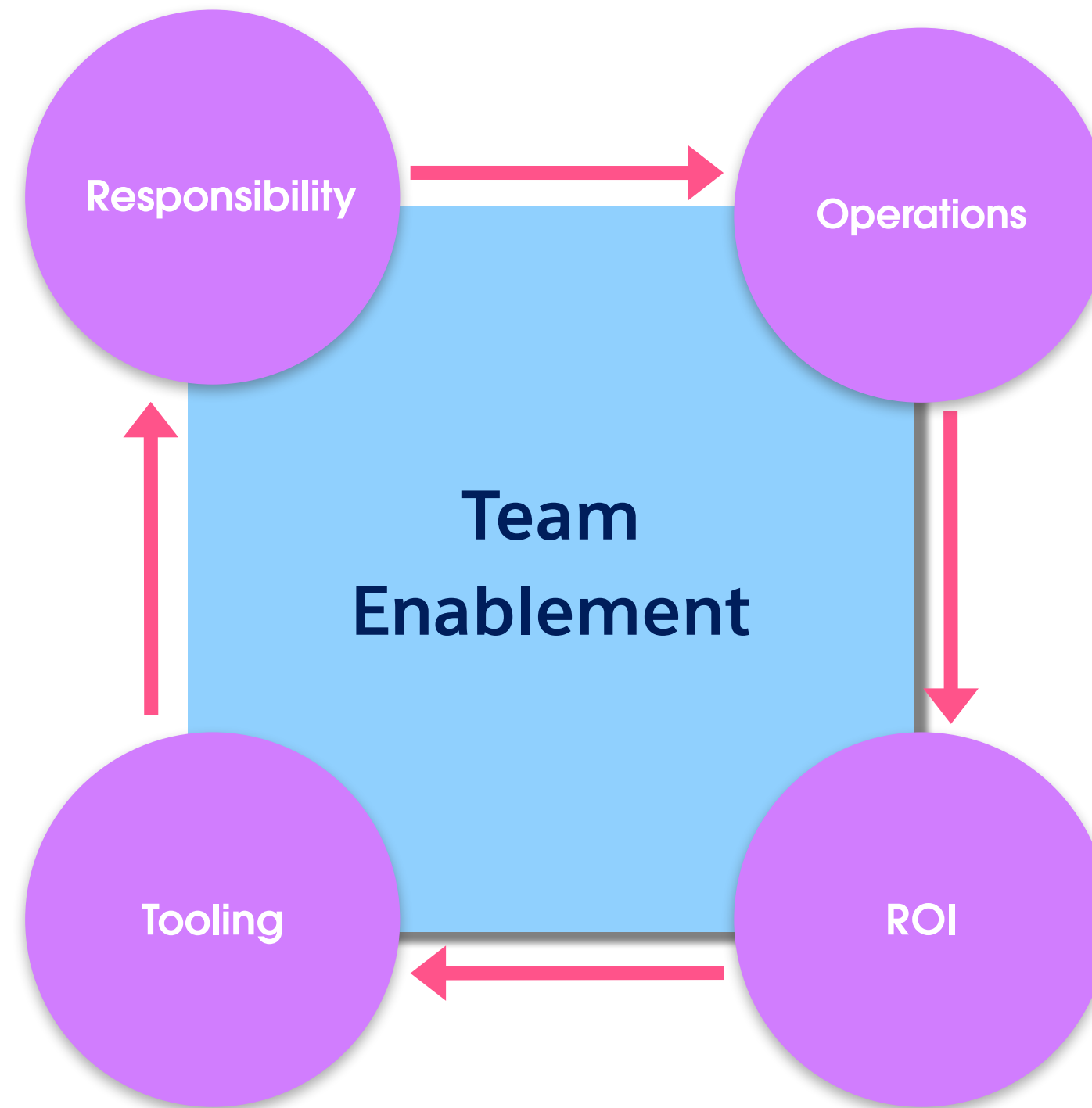
Lack of distributed responsibility

Shifting the entire DevOps responsibility to 1–2 people is a common antipattern, still causing bottlenecks in projects.

Wrong strategy for given project setup

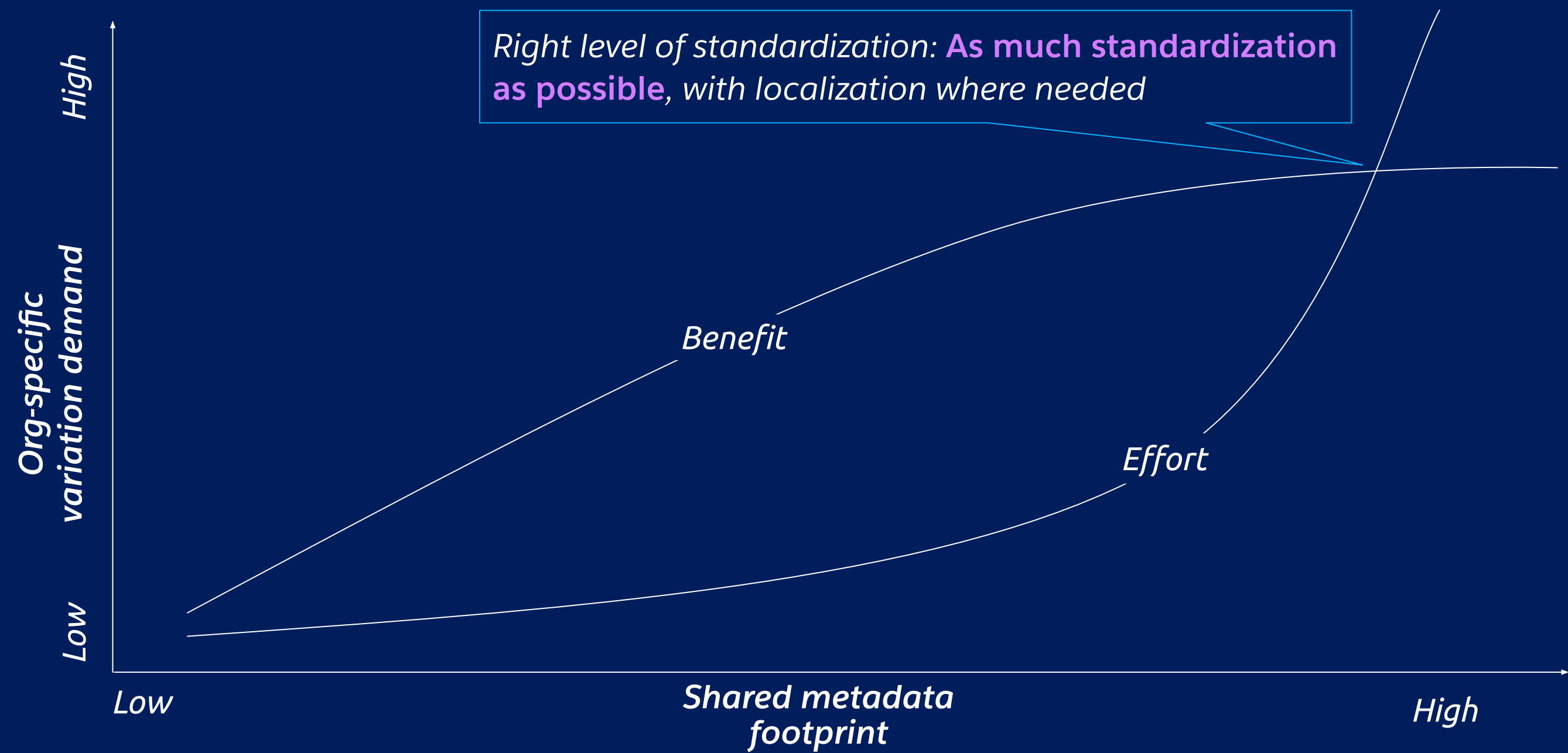
Delivery approach, org strategy and team size are all factors to be considered for choosing the right strategy.

Build a DevOps Culture That Fits Your Needs



A strong **DevOps culture** depends on team enablement

Specific Requirements and Legacy Systems Influence Degree of Standardization



There Are Two Common Approaches

Choose wisely!

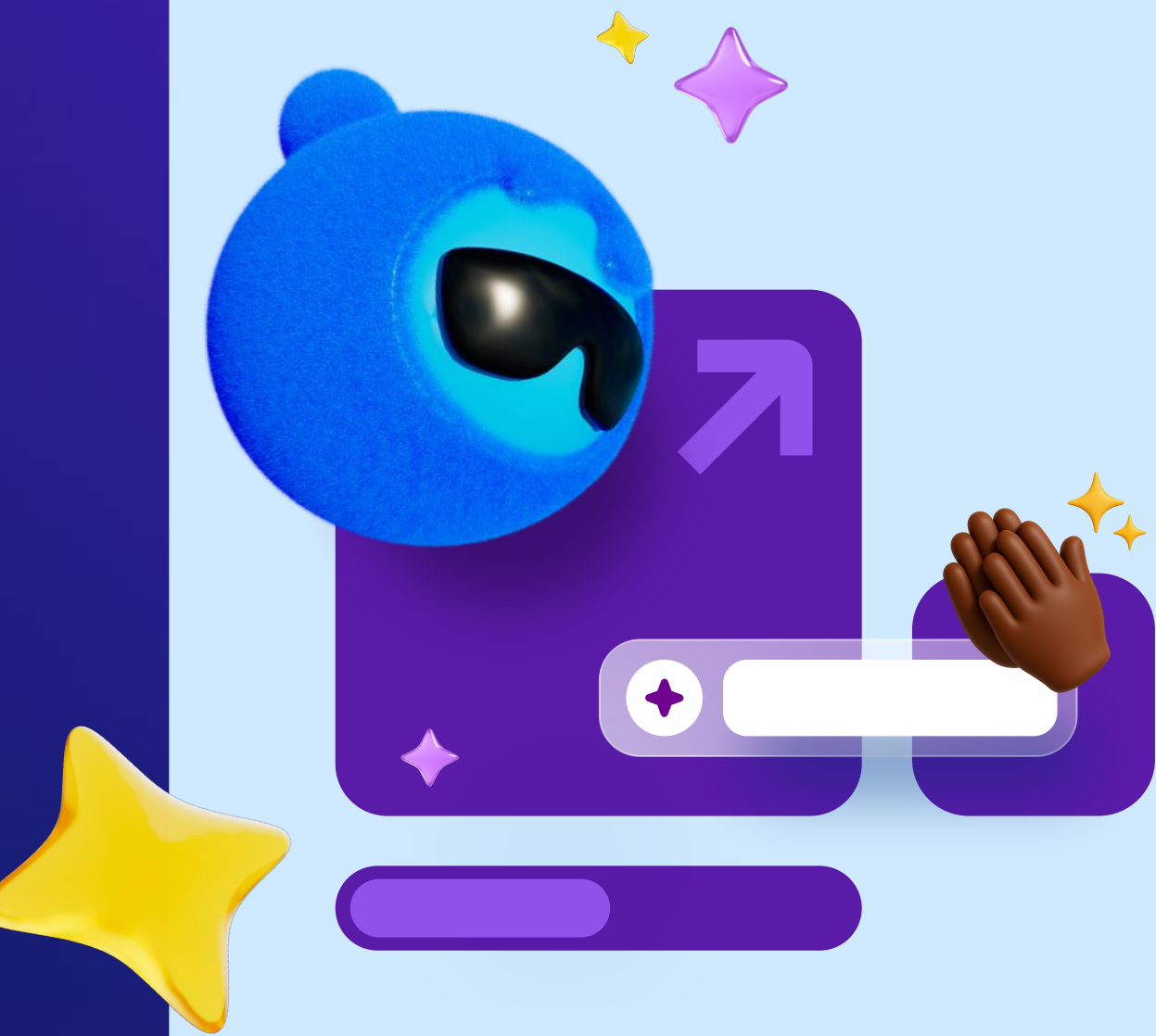


Package-Based Delivery

- Components versioned & distributed as unlocked/managed packages
- Each org installs a defined package version independently
- Package registry = single source of truth

Repository-Driven Delivery

- Git repo holds canonical metadata, pipelines deploy to target orgs
- Changes tracked via branches, PRs, and merge policies
- Repo is source of truth – not the org
- Maximum flexibility, but demands strong branching discipline



Package-Based Delivery

Have a Set of Common Metadata



Why Packages



The deployment unit that makes multi-org manageable

Versioned – every release is a defined snapshot; roll back to any prior version

Modular – decompose your org into logical boundaries (core, sales, service, integrations)

Dependency-aware – sfdx-project.json declares what depends on what;

Repeatable – same package version installs identically across 2 orgs or 20

Independently upgradeable – Org A can run v3.2 while Org B stays on v3.1

Governance-friendly – package versions create an audit trail of what was deployed

Complexity Management

salesforce

Three decisions you must get right before

Which packages are Org specific?

Separate what's shared from what's local – shared packages enforce consistency, org-specific packages preserve autonomy

Dependency Management

Map your package dependency tree before you build your pipeline, not after your first cross-org deployment fails

DevHub & Environment Strategy

Decide on one shared DevHub for visibility or per-BU DevHubs for autonomy – and always budget for a dedicated cross-org integration environment

Why Packages



Single-org

One org, one DevHub, one pipeline



Clean dependency chain – one version per package, one source of truth

Multi-org

3 orgs, diverging versions, coordination overhead

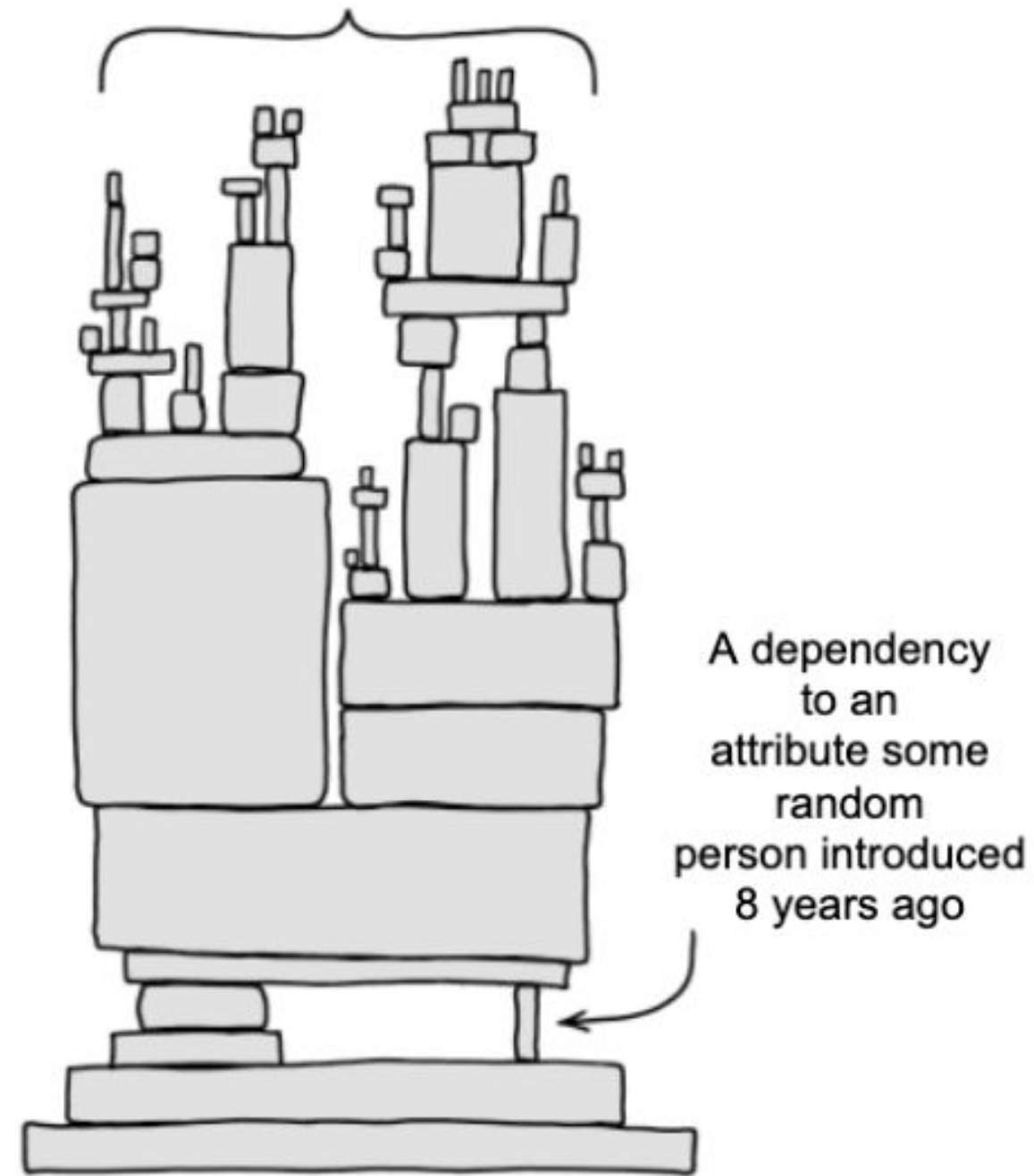


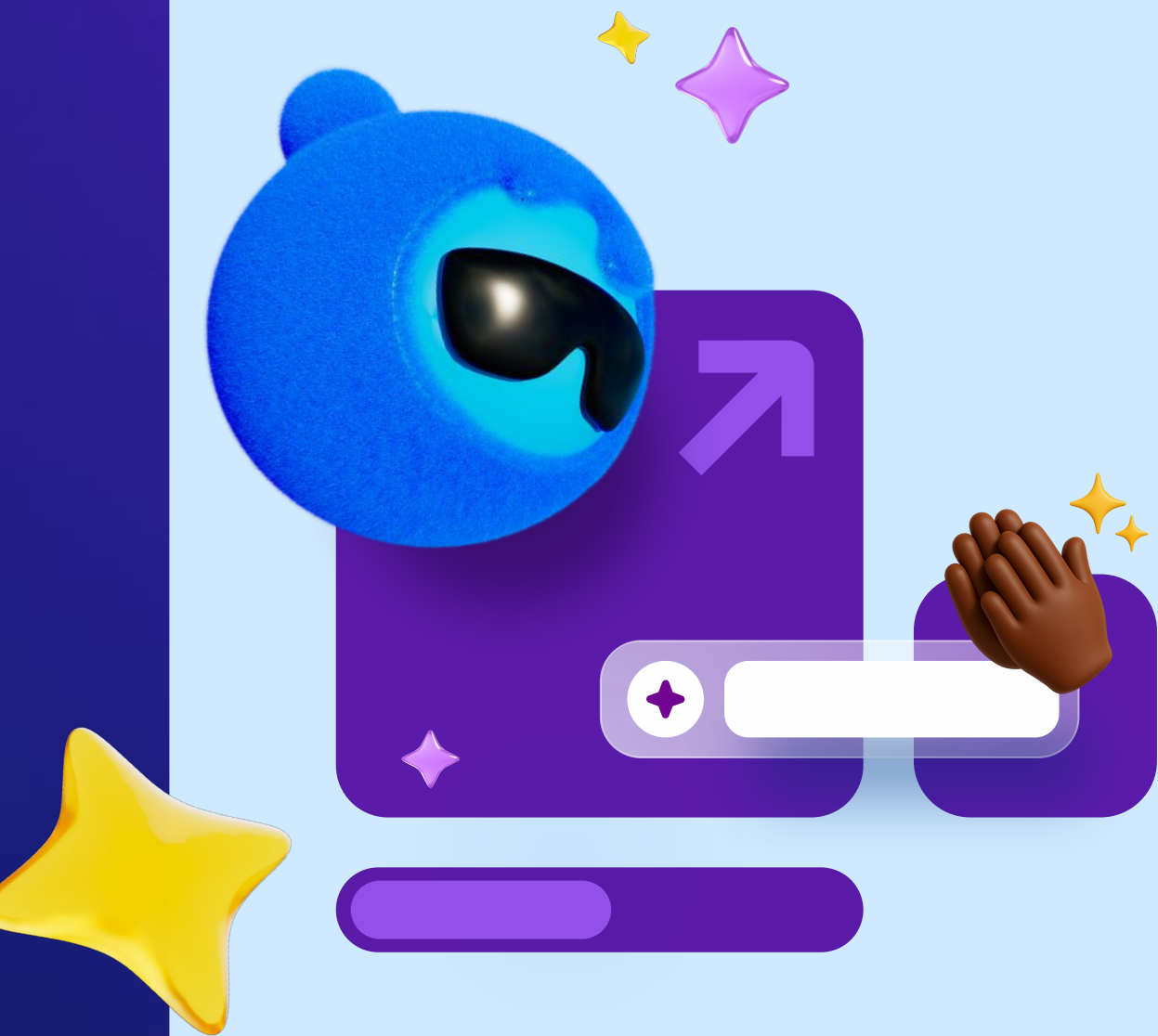
3 orgs × n packages × m versions = combinatorial explosion
Shared package upgrades must be coordinated across all orgs

We Learned It the Hard Way



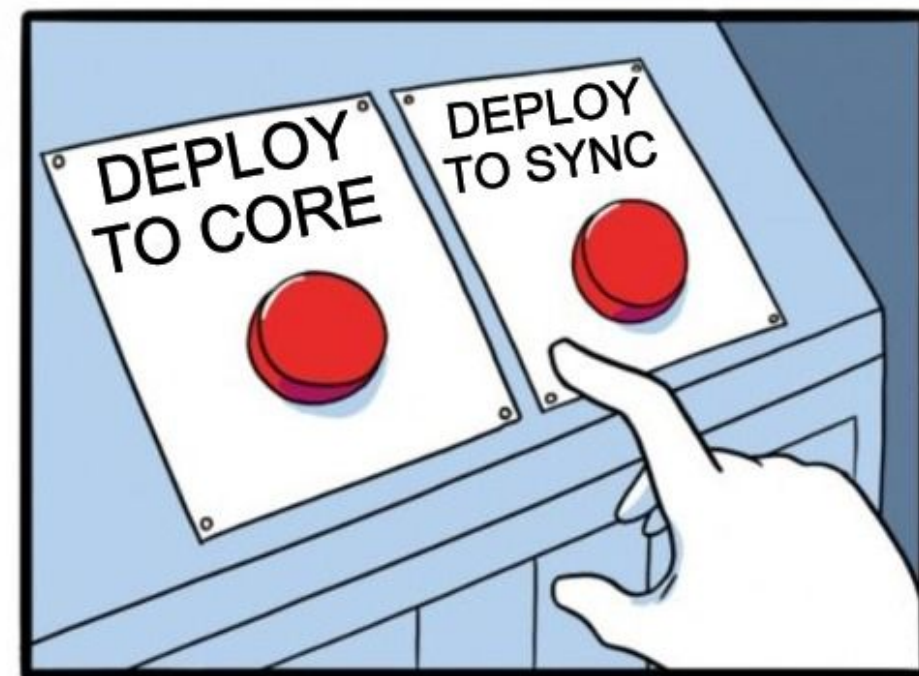
Our packing Approach





Repository-driven delivery

Have a set of common metadata



imgflip.com

JAKE-CLARK.TUMBLR

Why Repo?



Git as the single source of truth across all orgs

Canonical state – the repo defines what each org should look like

Diff deployments – deploy only what changed between commits, not full metadata snapshots

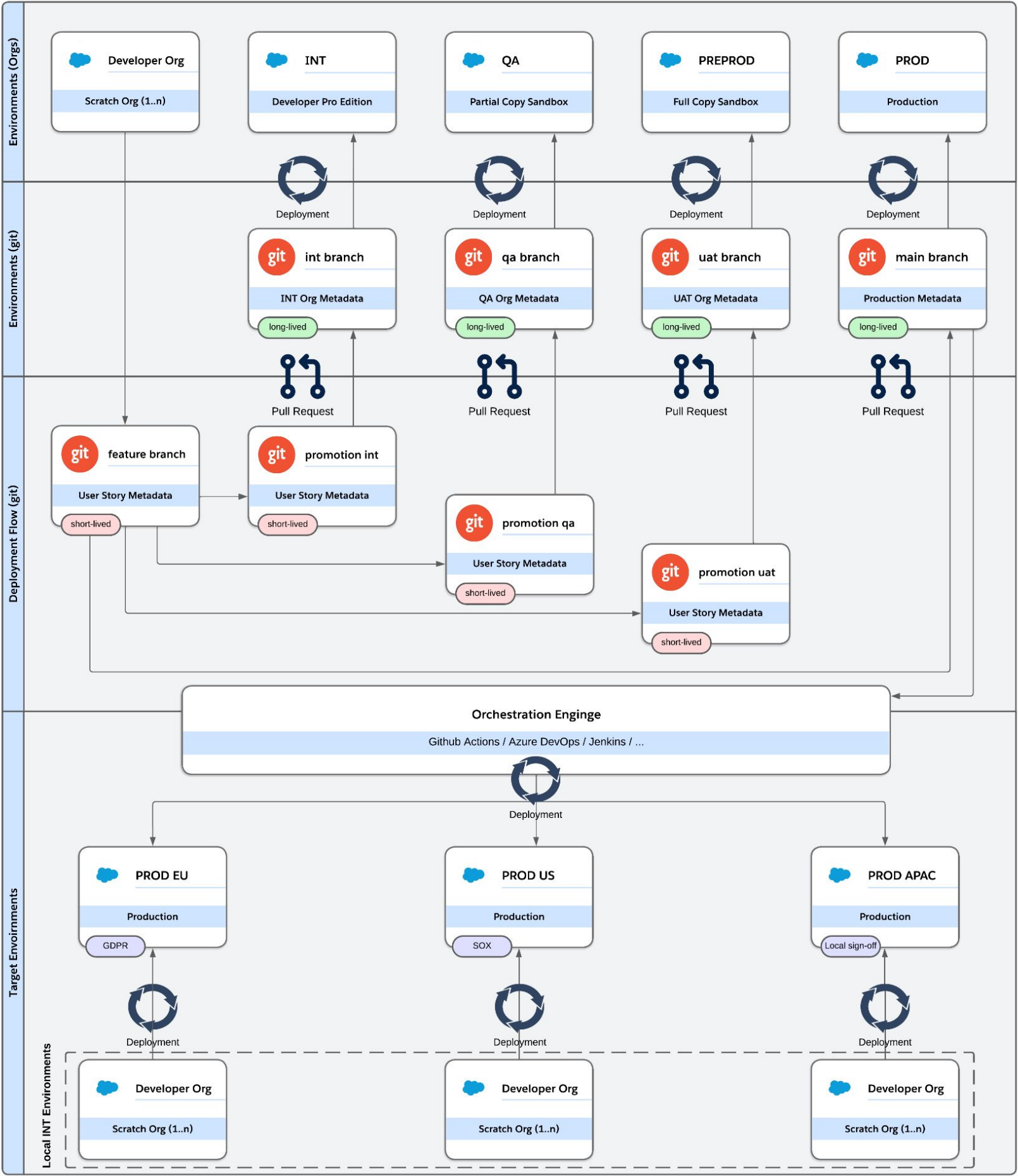
Branch-per-org – org-specific overrides live in dedicated branches or directories

Pull request governance – every change is reviewed and approved before it touches any org

Pipeline-as-code – deployment logic is versioned alongside metadata

Parallel promotion – promote the same commit to EU, US, APAC simultaneously

Pipeline Structure



Complexity Management

salesforce

Three decisions you must get right before

Monorepo or multi-repo?

One repo with directory-per-org gives central visibility and atomic cross-org commits

Multi-repo gives autonomy but sacrifices cross-org traceability.

Branch & merge strategy

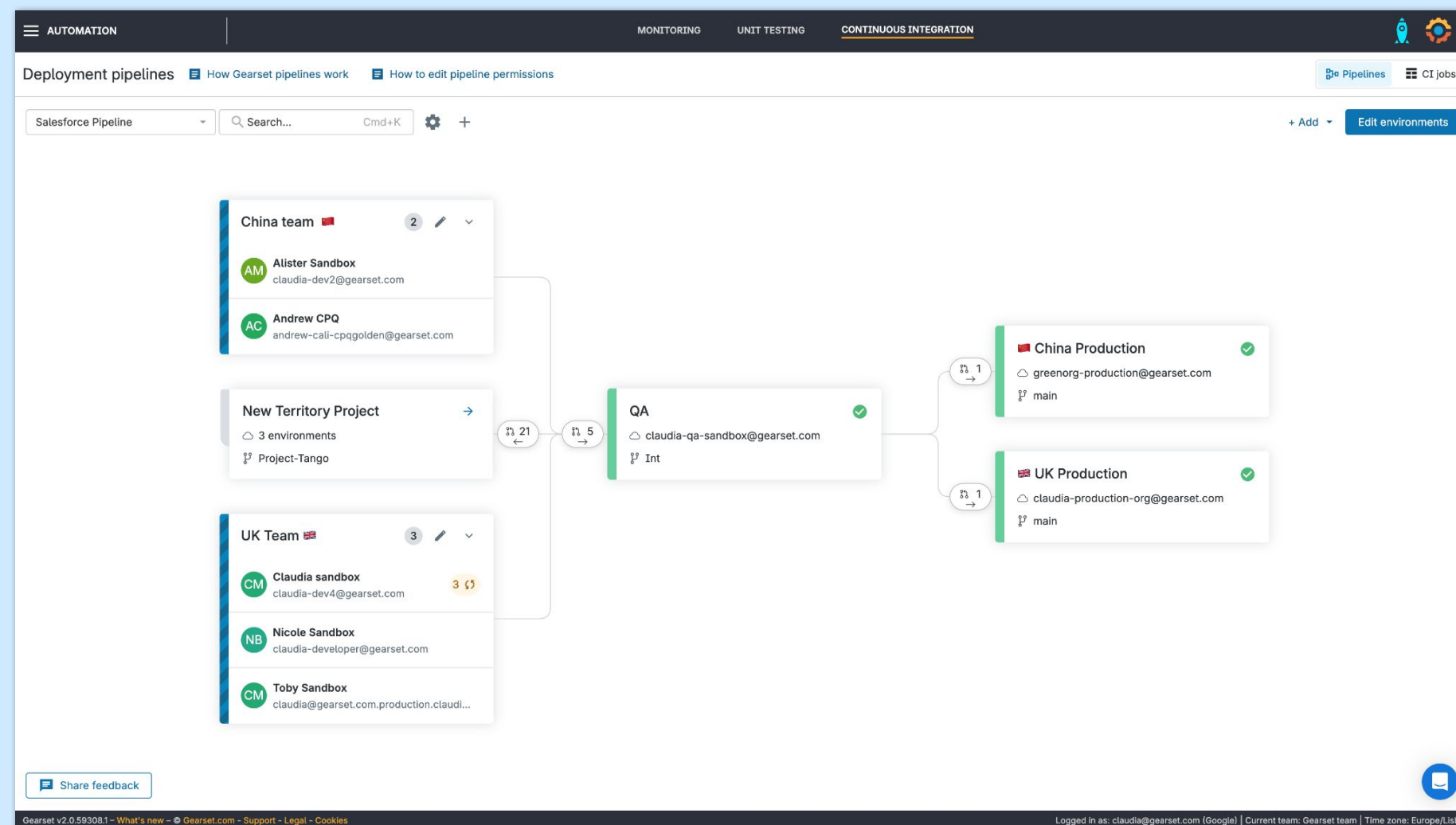
Define who merges what, when.
Without clear merge policies, your repo becomes the new source of drift instead of the cure.

Pipeline orchestration

A single **sf project** deploy won't cut it. You need a sequenced fan-out: shared components first, org deltas second, smoke tests third – with per-org rollback capability built in, not bolted on.

You don't have to build it all yourself

ISV partners offer DevOps tooling that
accelerates adoption

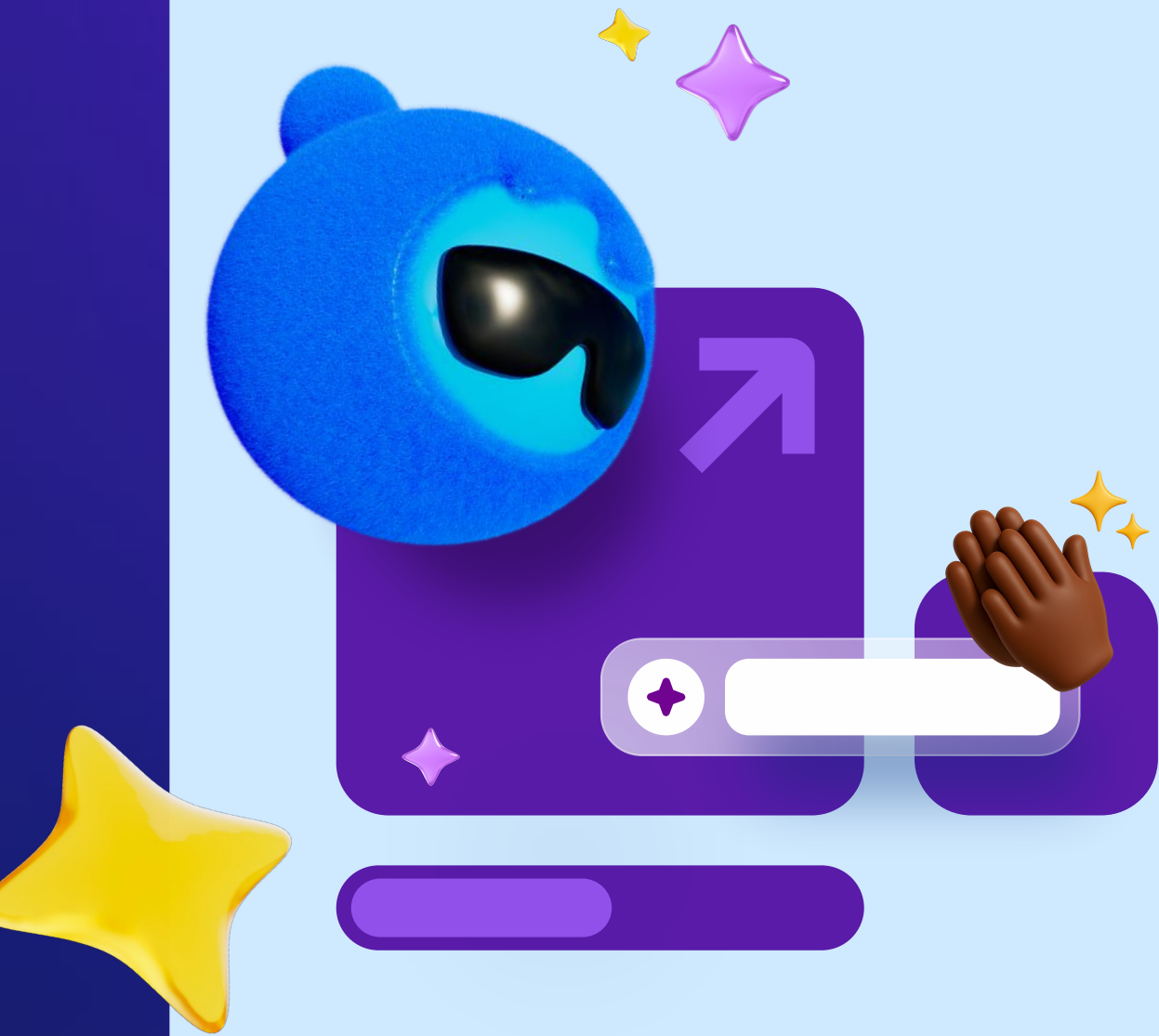


ISV Tool with Pipeline Management



We learned it the hard way





Our Experience

Multi-Org is not a Tooling-Issue. It's a Design Issue.





DEMO

If you remember nothing else from this session

salesforce

1

Start from the operating model, not the toolchain

Your org strategy is an enterprise architecture decision

2

The pipe between orgs is harder than the pipe within each org

Single-org DevOps patterns break at the org boundary

3

Make every architectural decision explicit — and document it

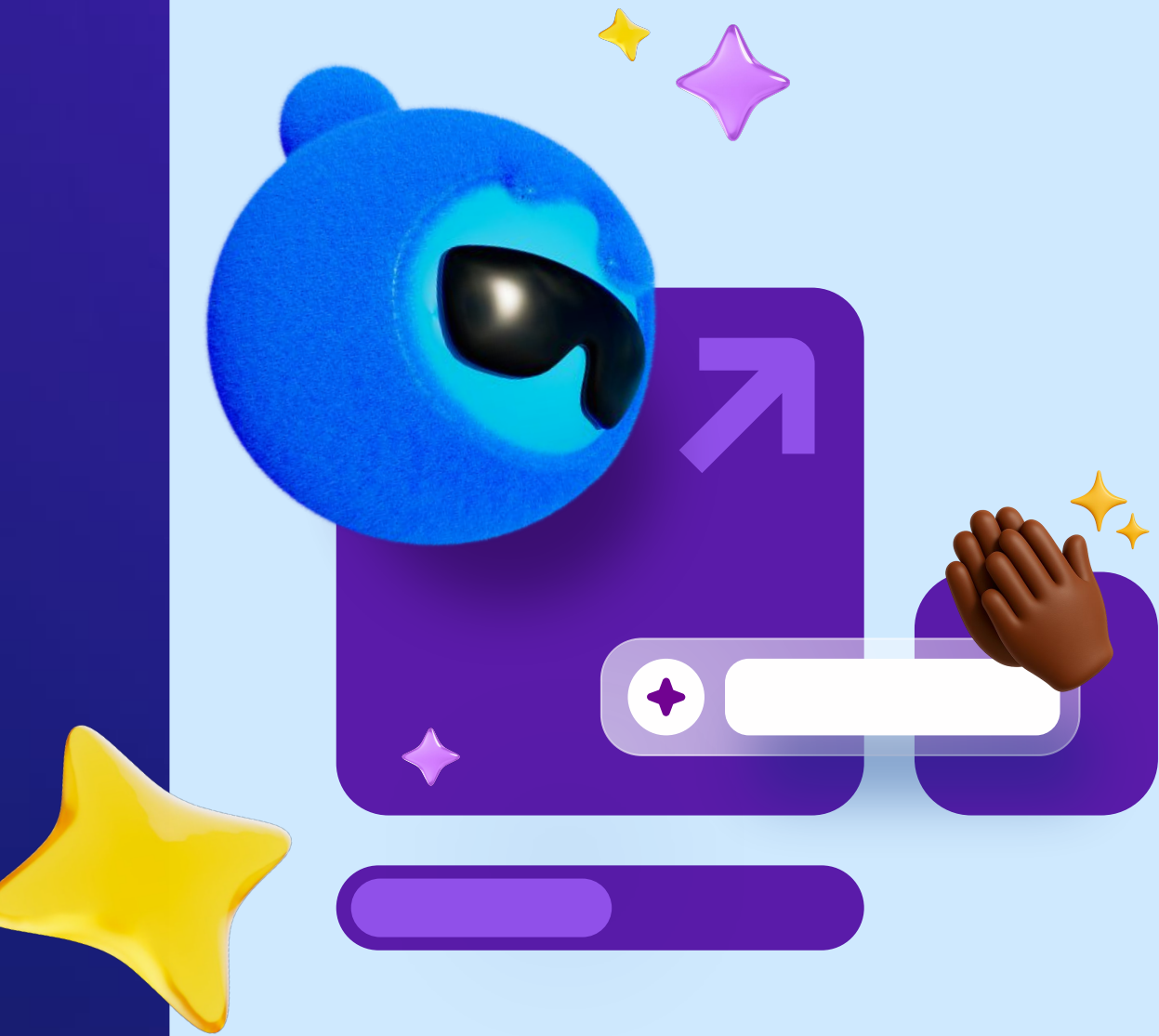
If these decisions live in someone's head instead of your repo, you don't have a strategy

4

Design for Agentforce and Data Cloud today

Agent definitions and D360 tenancy introduce cross-org dependencies that will grow

If your DevOps strategy
doesn't have a **multi-org
section**, you don't have a
DevOps strategy.



Q&A

Guiding principles for a package-based delivery approach



Package layer	Scope	Versioning	Upgrade cadence
Core platform Data model, security, base objects	Shared Installed in all orgs identically	Strict semantic versioning Breaking changes = major bump	Quarterly release train Aligned with SF releases
Integration layer Named creds, external services, events	Shared Contracts must be consistent cross-org	API-contract versioning Backward-compatible by default	On-demand + coordinated Triggered by API changes
Business logic Flows, Apex, automation, agents	Hybrid Shared base + org-specific extensions	Feature-branch per org Shared base locked; extensions float	Continuous for extensions Shared base follows core cadence
UI components LWC, Flexipages, app config	Org-specific Each org has its own UX needs	Org-level versioning No cross-org version sync needed	Continuous delivery Local teams release independently
Compliance & regulatory Audit trails, data residency, GDPR	Hybrid Global framework + local regulatory config	Immutable releases Audit trail per version mandatory	Policy-driven Deployed when regulation changes

Guiding principles for a repo-based delivery approach



Metadata layer	Repo structure	Branch strategy	Deployment trigger
Core platform Data model, security, base objects	<code>packages/core</code> Single source of truth across all orgs	Trunk-based on main No org-specific forks allowed	Merge to main Triggers deploy to all orgs sequentially
Integration layer Named creds, external services, events	<code>packages/integration</code> Tightly coupled with core	Trunk-based, pinned to API versions Contract-first approach	After core, before org deltas Order enforced by pipeline
Business logic Flows, Apex, automation, agents	<code>packages/logic + orgs/{region}/logic</code> Shared base + org overrides	Feature branches → main Org branches for overrides	Shared: with core cadence Overrides: continuous per org
UI components LWC, Flexipages, app config	<code>orgs/{region}/ui</code> Fully org-specific directories	Long-lived org branches Or org-scoped dirs on main	Continuous delivery Local teams push independently
Compliance & regulatory Audit trails, data residency, GDPR	<code>packages/compliance + orgs/{region}/</code> Global framework + local config	Protected branch Compliance team approval on every PR	Policy-driven Deployed when regulation changes



GITHUB REPOSITORY

