

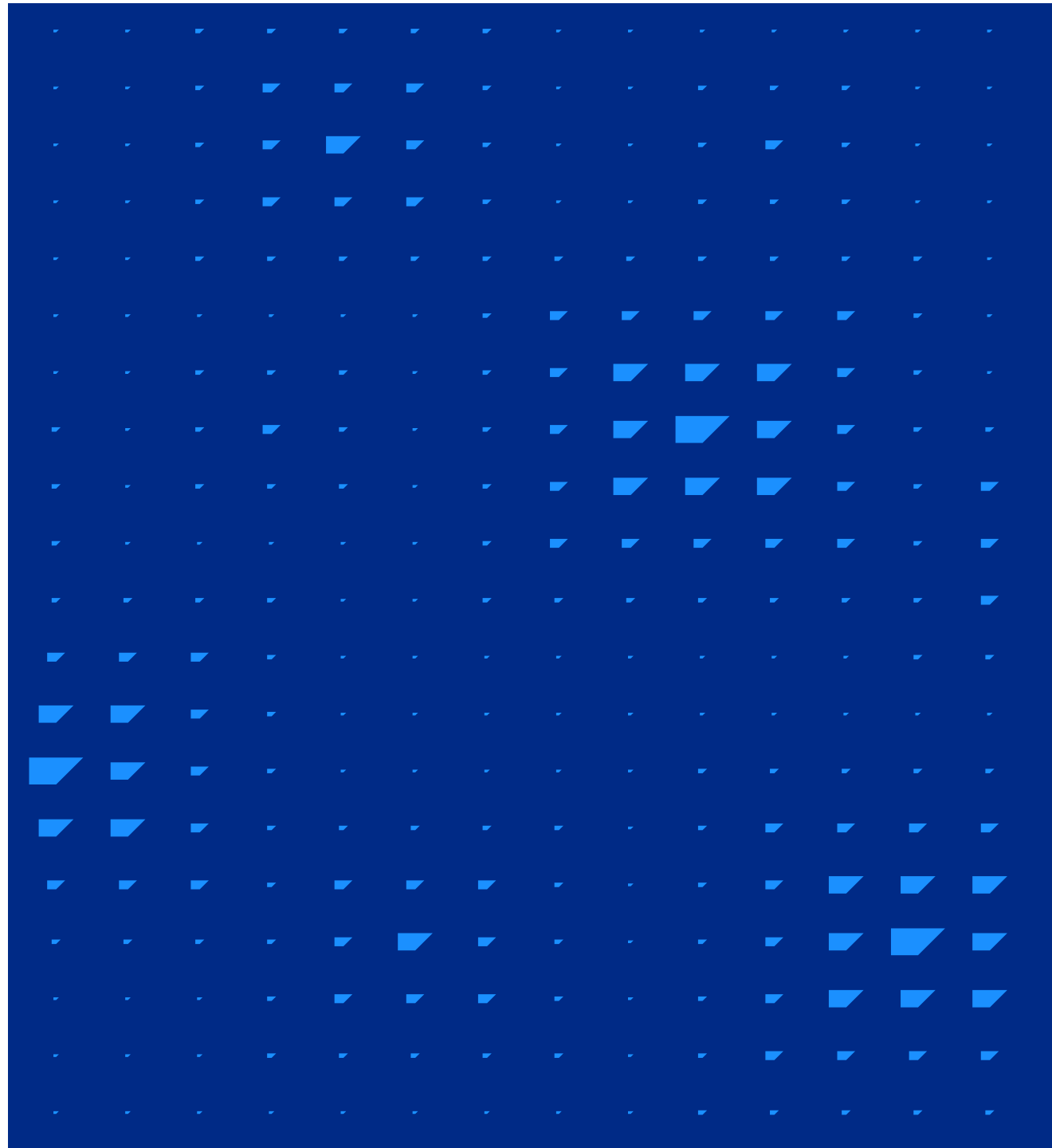


Platform Engineering on SAP BTP

Make the Life of your Developers easier

Christian Lechner, SAP

September 19, 2026





Hi, I am Christian

Product Manager Platform Automation (btp CLI & Terraform)

❤️ Developer Experience

E: christian.lechner@sap.com

L: <https://www.linkedin.com/in/christian-lechner-inthecloud/>

Disclaimer

The information in this presentation is confidential and proprietary to SAP and may not be disclosed without the permission of SAP. Except for your obligation to protect confidential information, this presentation is not subject to your license agreement or any other service or subscription agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or any related document, or to develop or release any functionality mentioned therein.

This presentation, or any related document and SAP's strategy and possible future developments, products and or platforms directions and functionality are all subject to change and may be changed by SAP at any time for any reason without notice. The information in this presentation is not a commitment, promise or legal obligation to deliver any material, code or functionality. This presentation is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This presentation is for informational purposes and may not be incorporated into a contract. SAP assumes no responsibility for errors or omissions in this presentation, except if such damages were caused by SAP's intentional or gross negligence.

All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.

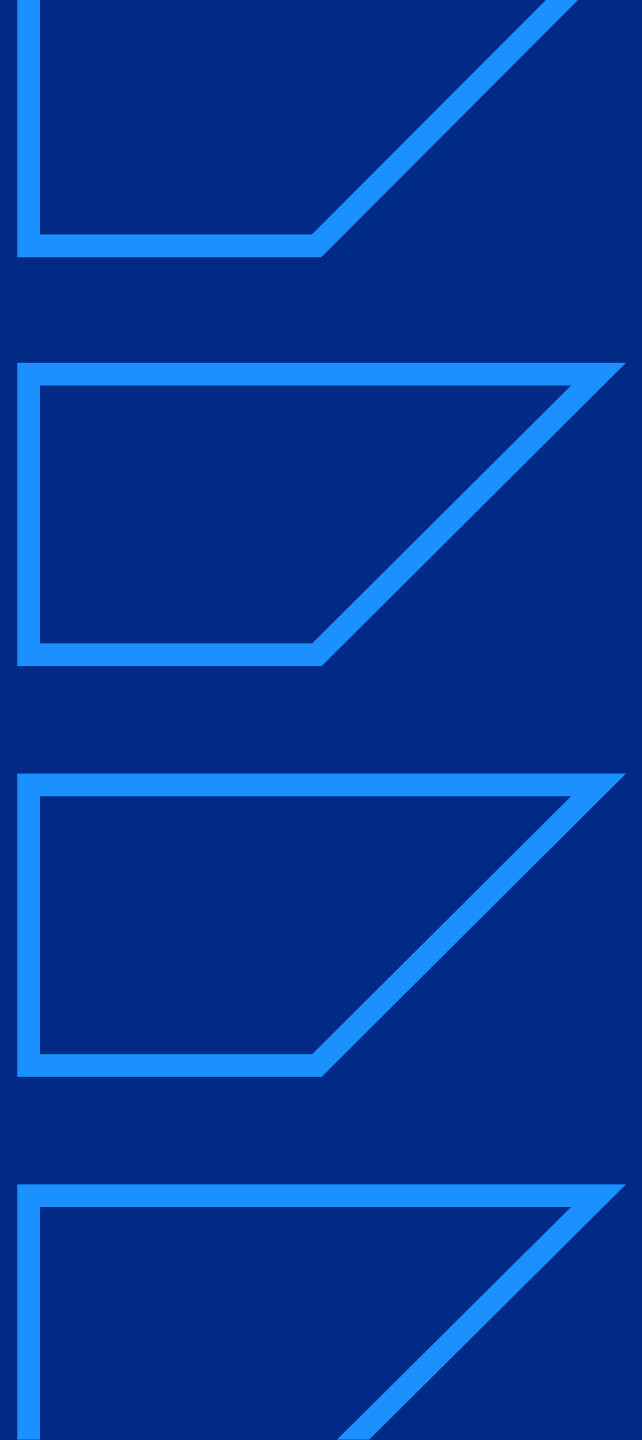
Agenda

01 What about Infrastructure Automation?

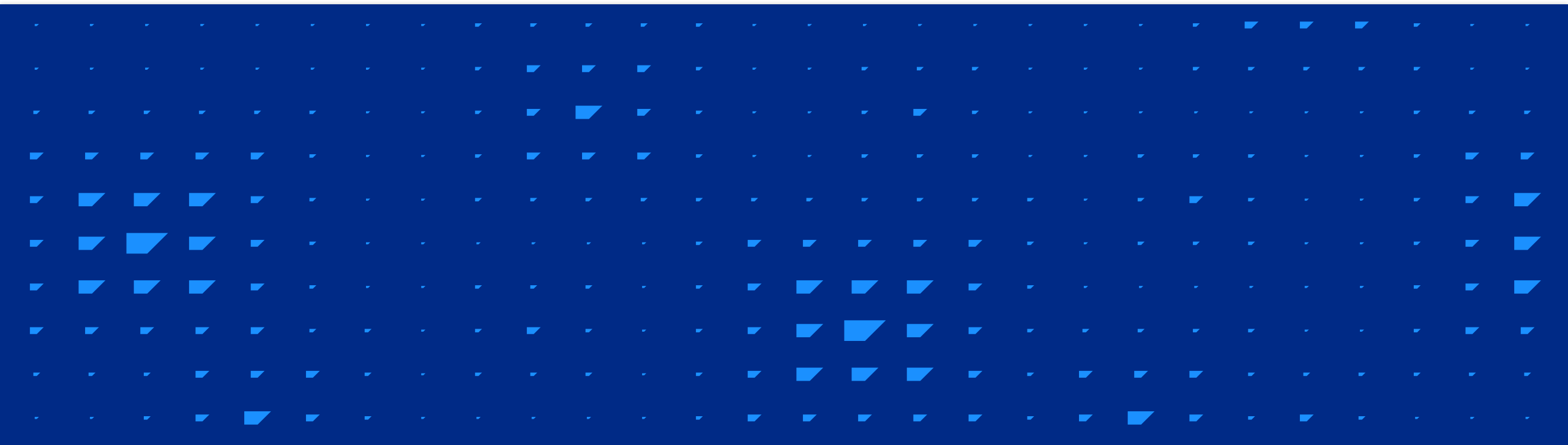
02 What about Developers?

03 How to improve the Life of Developers?

04 What about AI?

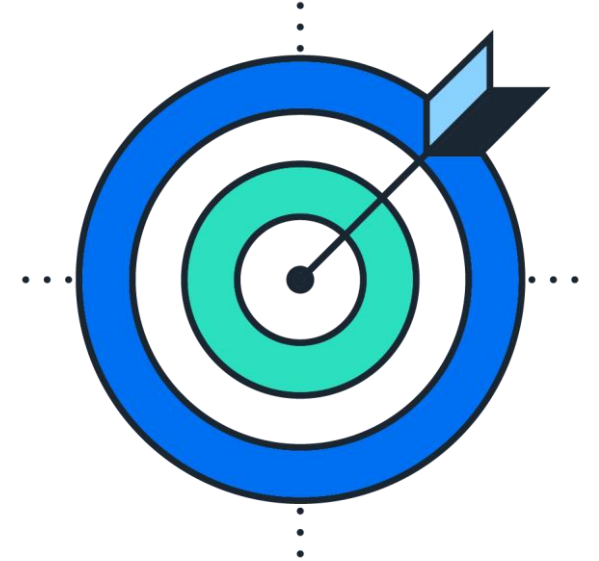


What about Infrastructure Automation?

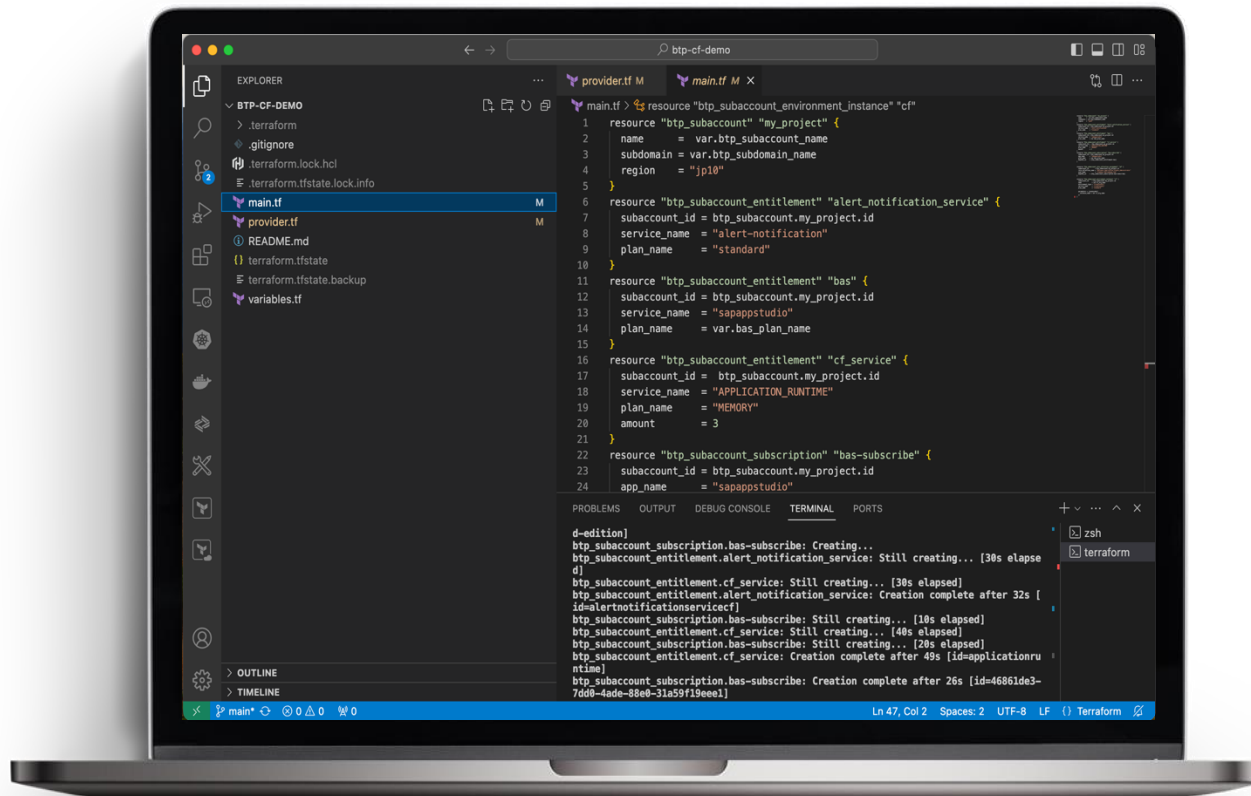


Motivation

1. Find the most efficient approach to *provision* and *manage infrastructure* in a *compliant way*
2. Use current (de-facto) *industry standards*
3. Rely on a *mature* and *established ecosystem* for knowledge and tooling



Infrastructure as Code via Terraform



Terraform as IaC tool enables us to **provision** and **manage** of infrastructure **through code** instead of through manual processes.

Infrastructure is defined **declaratively**

Central storage of **state**

On top

- Enablement of a consistent ***governance*** setup when ***managing infrastructure*** on ***SAP BTP***
- ***Baseline*** for ***audits*** of ***provisioning*** and ***management*** of infrastructure
- Ensuring ***secure setups*** via automation
- Creation of the baseline for ***FinOps***
- ***Easier*** and ***faster*** move between landscapes (sovereign cloud)



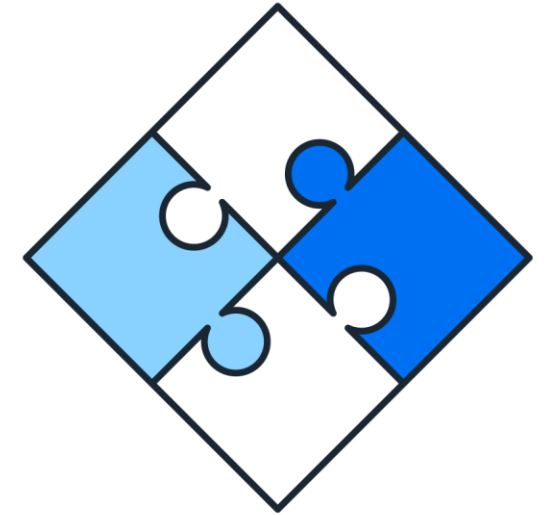
But this requires ...

Deep knowledge of the SAP BTP as **infrastructure** layer:

- What entitlements are available?
- Which service plans and what do they cost?
- ...

Knowledge about **organizational constraints** for the **infrastructure**:

- Naming conventions for subaccount etc.
- Labeling conventions
- ...

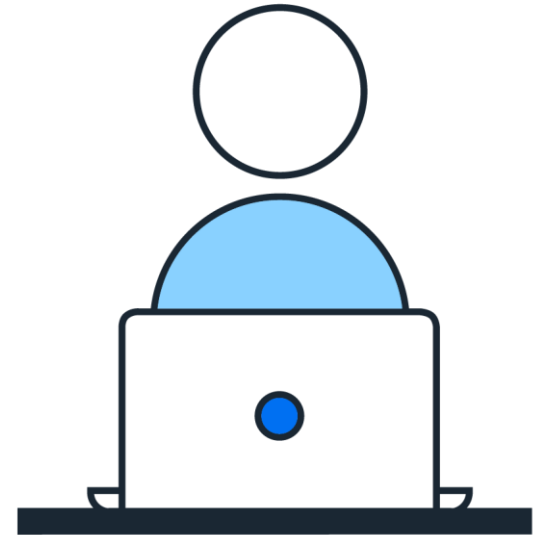


Covered by ...

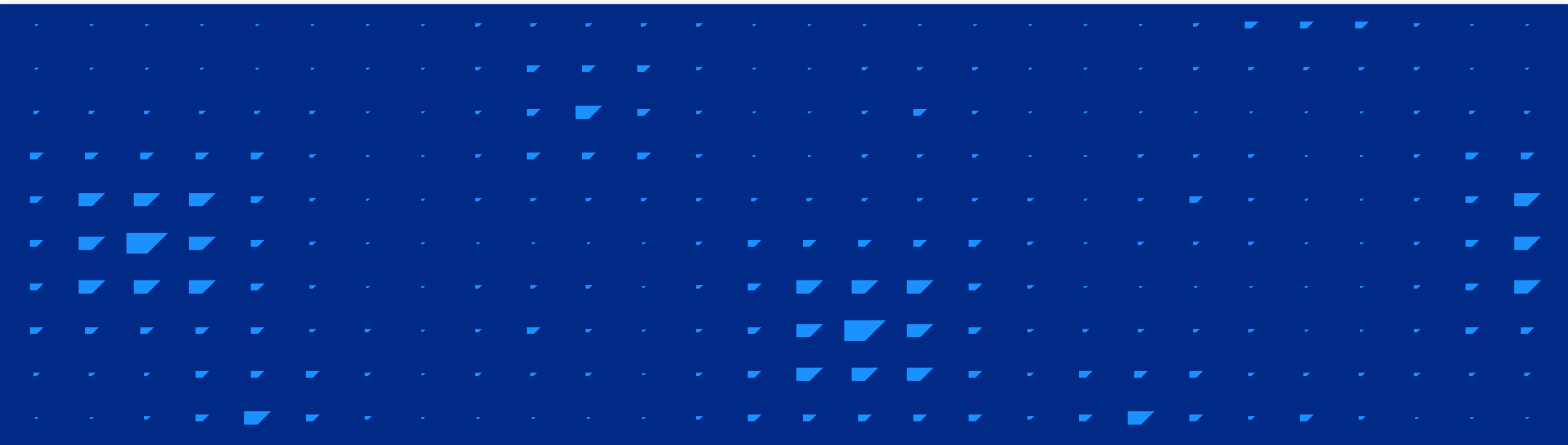
Administrators

Platform Engineering

DevOps Team



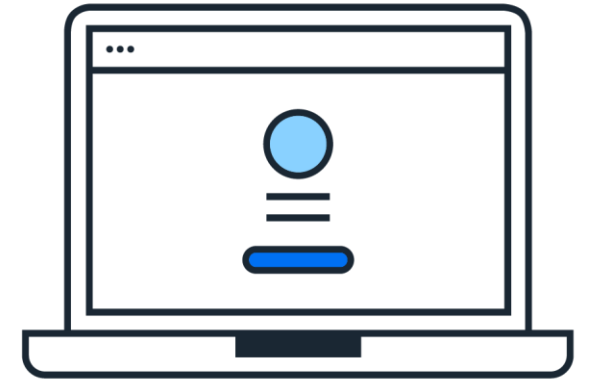
What about Developers?



What is the job of developers?

Development teams **build business applications** following the **clean core principles**

➔ **Enable** the **business departments** to do their job



But ...

They should **not** need to care about SAP BTP as **infrastructure** layer and the **organizational constraints** for the **infrastructure**



HOW TO

BRIDGE THE GAP?

imgflip.com

Baseline

Align established setups on SAP BTP

- Account structure
- “Allowed” services and applications including plans
- Shared services
- What information is needed to do proper FinOps
- ...

Translate established setups into **codified blueprints**
via **Terraform** (by Administrators)

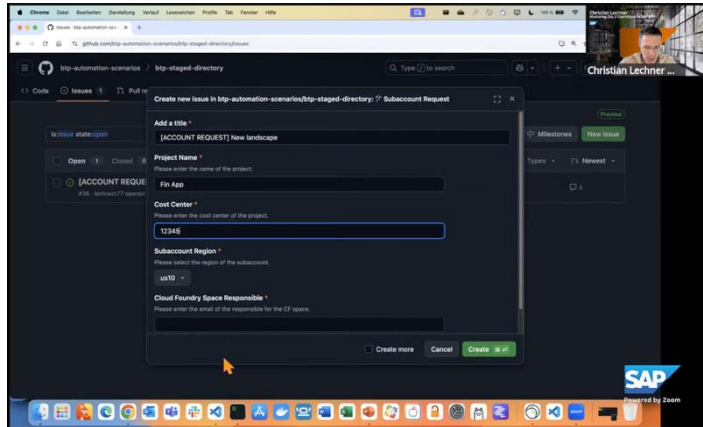
What's next?

Do not hand over blueprints via a *curated Terraform module catalog* to *development teams*

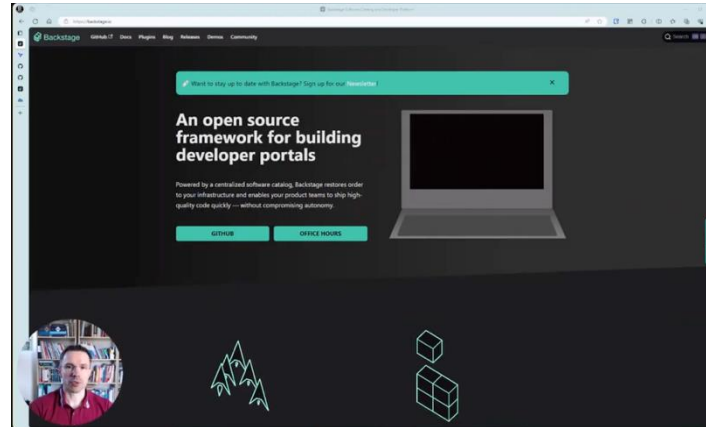
BUT: Provide *self-services* for the development teams to *provision these setups*

Self Service using Terraform

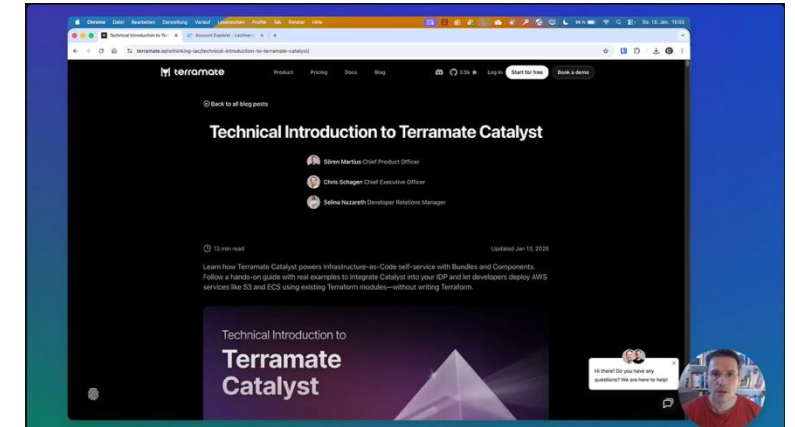
You can already do this today using the ecosystem around Terraform



Via GitHub PRs
and Actions
([Link](#))



Via Backstage
([Link](#))



Via Terramate Catalyst
([Link](#))

Terramate Catalyst Demo - Assumptions

We have established different blueprints for app development

Development team starts a new project relying on one of these blueprints

Operations team offers a self-service tool for the infrastructure setup

All project requests are issues in a central repository

Flow

Step 1: The development team opens an issue in the central repository
→ a project-specific repository for the infrastructure code gets created
(owned by Admin team)

Step 2: An issue gets automatically created in the project repository and assigned to the requestor of the development team

Step 3: The requestor uses a CLI tool and creates a machine-readable project description based on the curated content and creates a pull request

Step 4: The PR gets reviewed and merged by admin team

Step 5: The automated setup creates the infrastructure

Step 6: The development team gets informed (e.g., via Teams) once the setup is done

Flow

Step 1: The development team opens an issue there
→ a project-specific repository for the infrastructure code gets created
(owned by Admin Team)

Step 2: An issue gets automatically created in the project repository and assigned to the requestor of the development team

Step 3: The requestor uses a CLI tool and creates a machine-readable project description based on the curated content and creates a pull request

Step 4: The PR gets reviewed and merged (ops teams)

Step 5: The automated setup creates the infrastructure

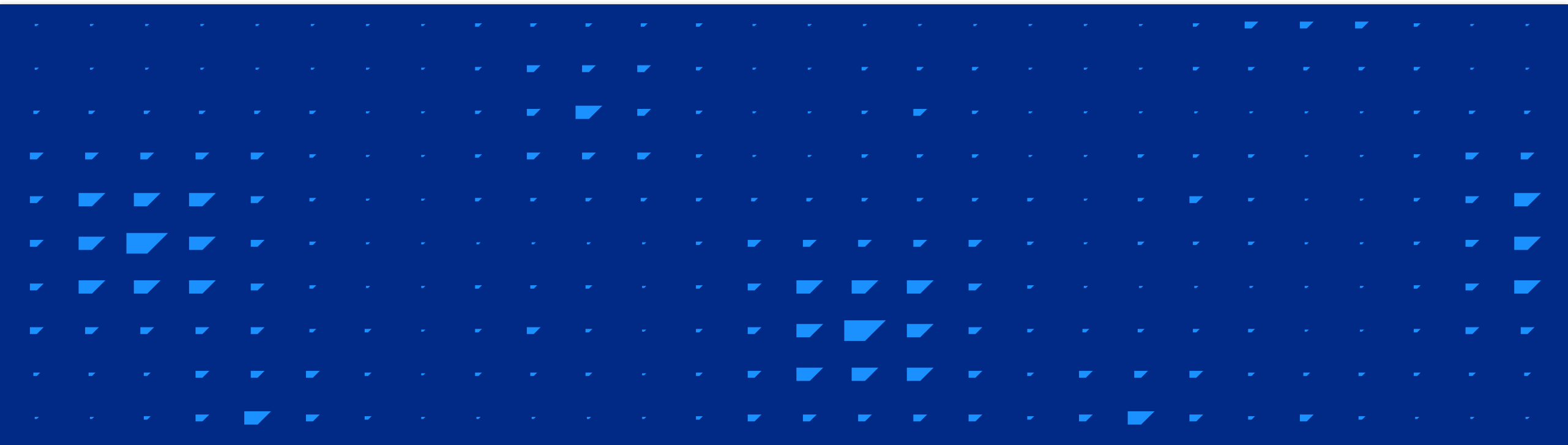
Step 6: The development team gets informed (e.g, via Teams) once the setup is done

DISCLAIMER

We will do all process steps locally for the sake of transparency

All provisioning steps can be automated (with a human in the loop)

Demo – Terramate Catalyst



EVERYBODY GETS



AI AND AGENTS

imgflip.com

Before we even start ...

LLMs are fundamentally **probabilistic**, autoregressive **models** that **predict the most likely next token** (word or sub-word). They generate **non-deterministic outputs** by sampling from probability distributions over their vocabulary.

Halucinations are an integral part of the architecture

→ **Keeping this in mind, they can be really useful!**

Prerequisite

Ensure that:

- **deterministic guardrails**
(e.g. tflint, Open Policy Agent)
- **automated tests** (terraform test)
- **monitoring**

are in place

Terraform Code

Tools like Claude Code are already good in writing and reviewing Terraform code

Use **Model Context Protocol** (MCP) Servers

- <https://github.com/hashicorp/terraform-mcp-server>
- <https://github.com/opentofu/opentofu-mcp-server>

Use **Agent skills**

- <https://github.com/hashicorp/agent-skills>
- <https://github.com/antonbabenko/terraform-skill>

Provide **organization specifics, guidelines, best practices and examples** via linting rules and/or markdown files

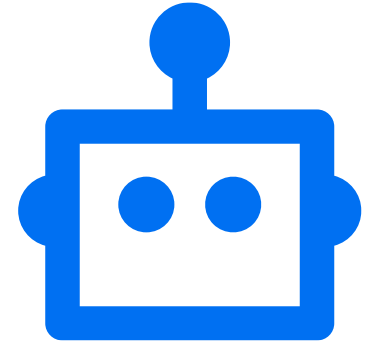


Spec-driven Development for IaC

Define **medium and complex** setups via a spec-kit

Provide a **defined flow** for **crafting the specification(s)**

Get **guidance** and **good practices baked in** the setup



tl;dr

Terraform is a powerful tool for provisioning and managing Infrastructure as Code

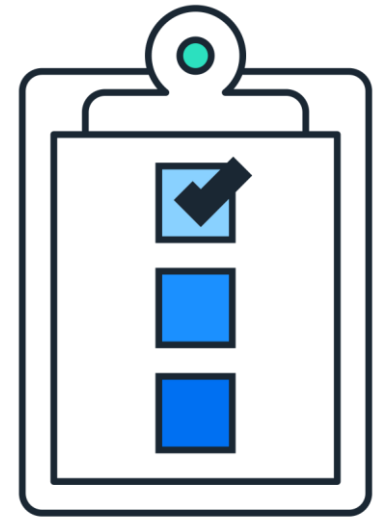
Development Teams require **curated infrastructure setups** to fulfill their jobs = developing and delivering business applications

Don't bother development teams with knowledge about the **infrastructure details** of **SAP BTP**

Make the life of **developers** easy, by providing **self services** for setups on SAP BTP leveraging **Terraform**

Technical things can be **solved** (easily), but **organizational prerequisites** also need to be put **in place (usually harder)**

AI and **Agents** can **help you** in your journey, **spec-driven development** makes it even better 😊

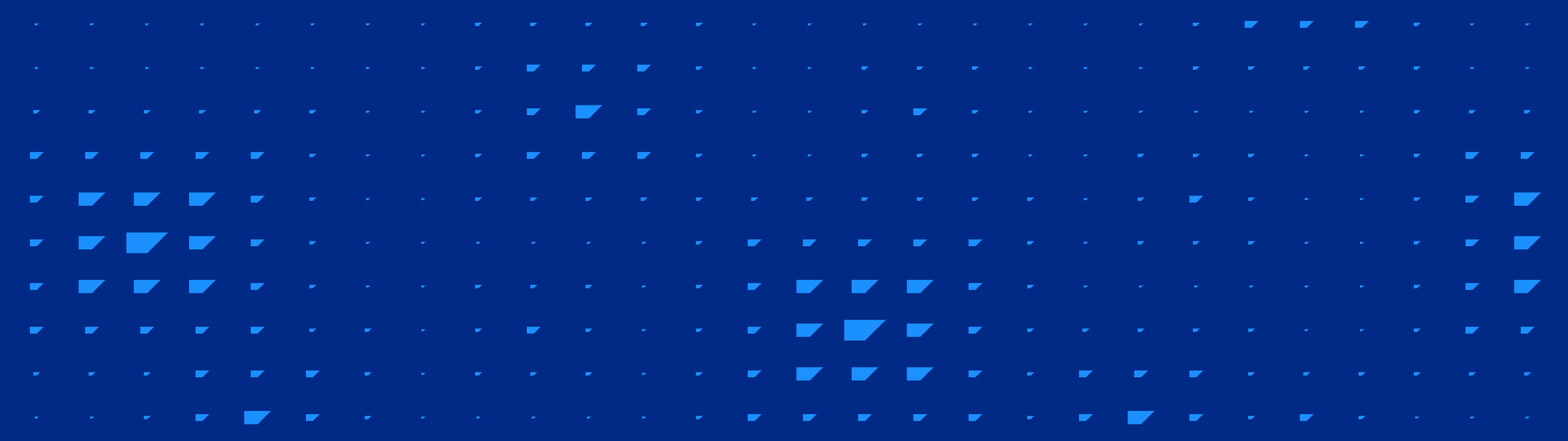


Your perspective matters

Join the conversation

You ask. We answer.





Thank you.

Christian Lechner
Product Manager

christian.lechner@sap.com