SOFTWARE ARCHITECTURE DECISIONS: THE ASSURE JOURNEY

AITOR ECHEVARRÍA FERNÁNDEZ

RUBÉN LÓPEZ APARICIO

MARCH 2025

INTRO

Aitor Echevarría

Software Architect

DXC Technology

Rubén López

Software Architect

DXC Technology

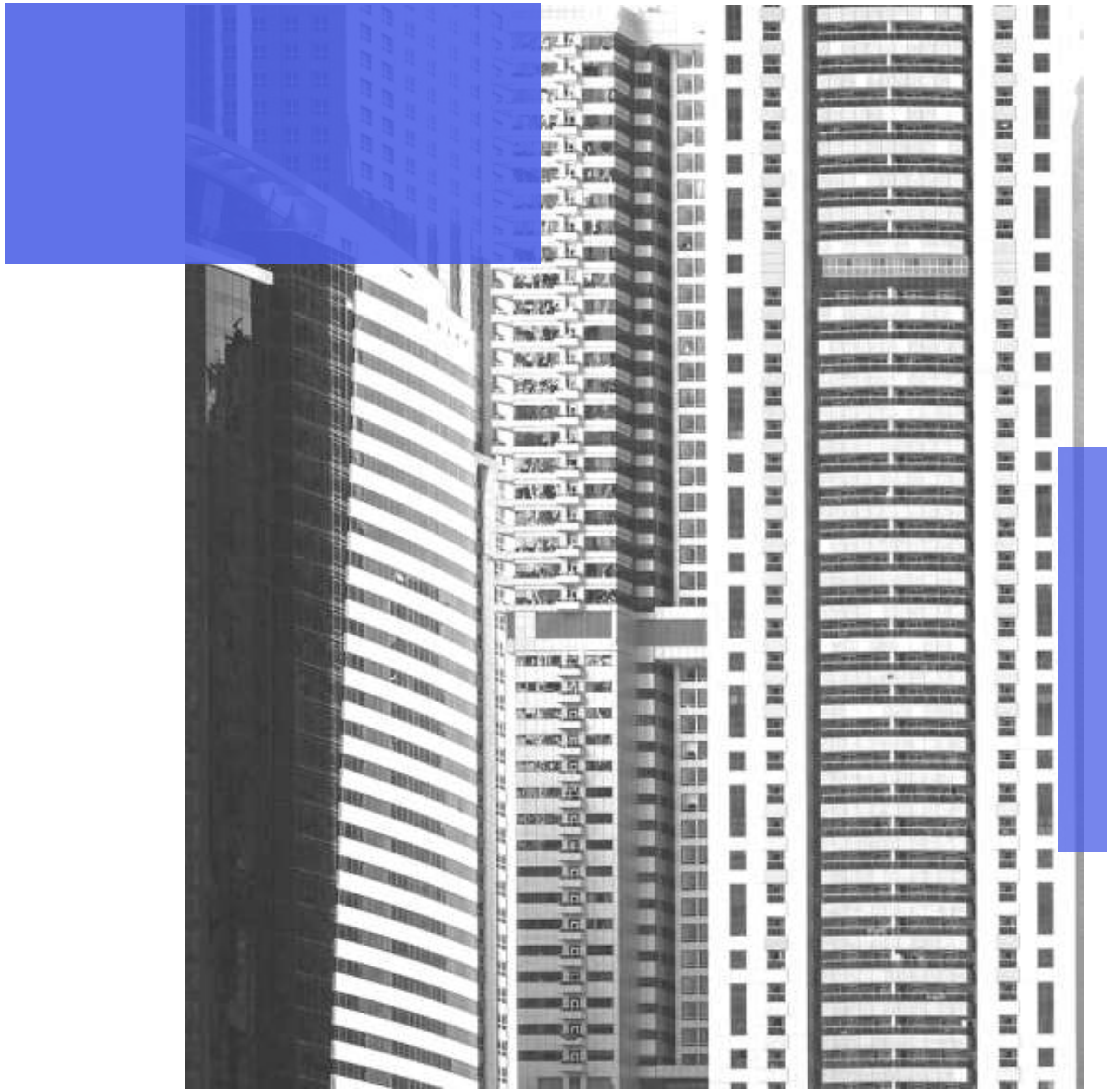
WHY TWO SPEAKERS?

> “EVERYTHING FAILS ALL THE TIME. THE BEST WAY TO AVOID FAILURE IS TO EMBRACE FAILURE AS A NATURAL OCCURRENCE AND BUILD RESILIENT SYSTEMS THAT RECOVER AUTOMATICALLY”

Werner Vogels (AWS CTO)

◈ But **this presentation is “stateful”** (keeping the analogy, Aitor and Rubén have distributed the contents to present exclusively). Is this pattern still valid for this presentation? Which are the challenges?

✓ **Failover pattern** - ensures that a system remains operational even if a component fails switching to a backup component.



AGENDA

UNDERSTANDING SOFTWARE ARCHITECTURE

CASE STUDY: THE ASSURE JOURNEY

PRESENT AND FUTURE OF SOFTWARE ARCHITECTURE

EXPECTATIONS VS REALITY

Aitor Echevarría

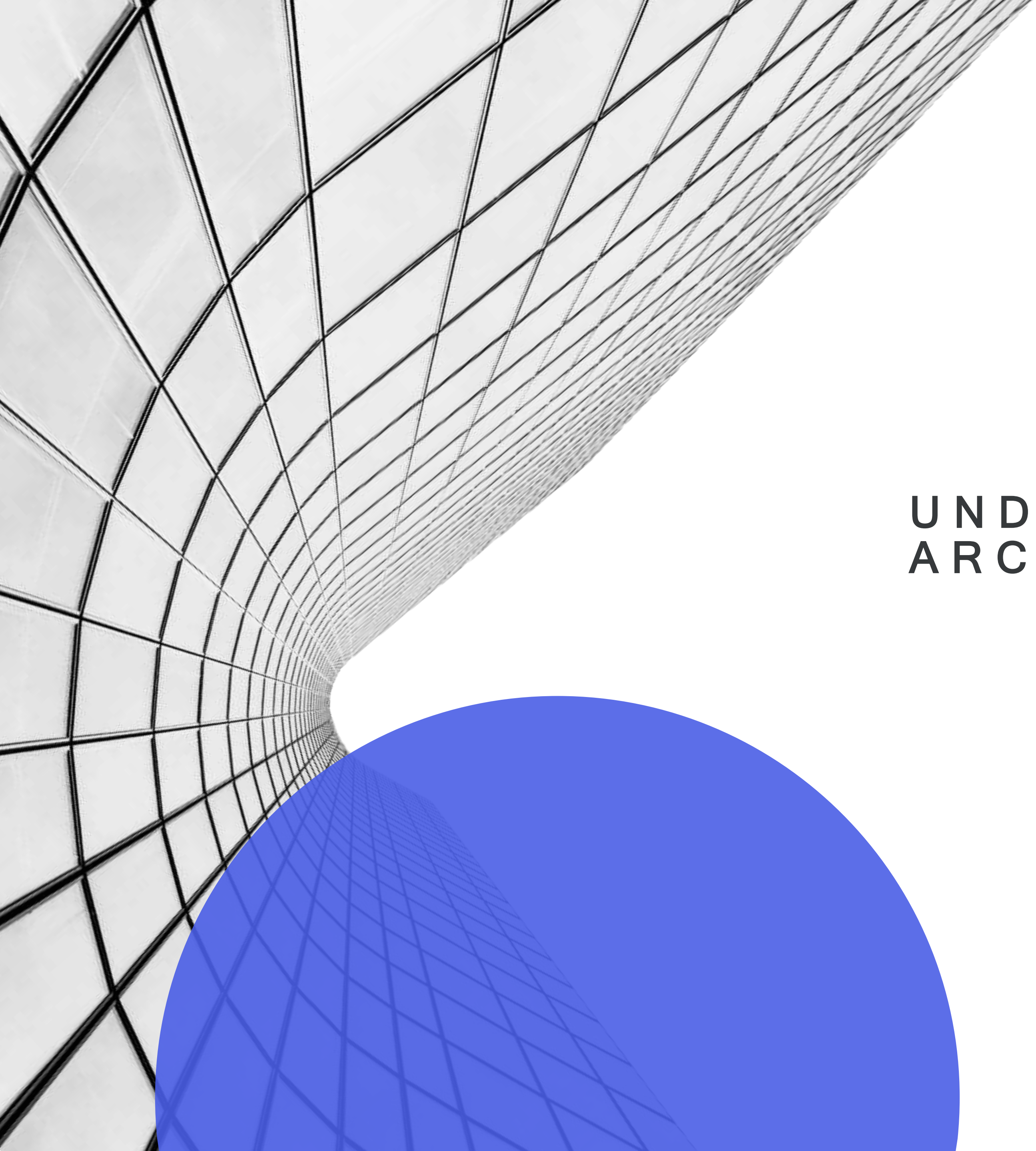
Software Architect

DXC Technology

Rubén López

Software Architect

DXC Technology

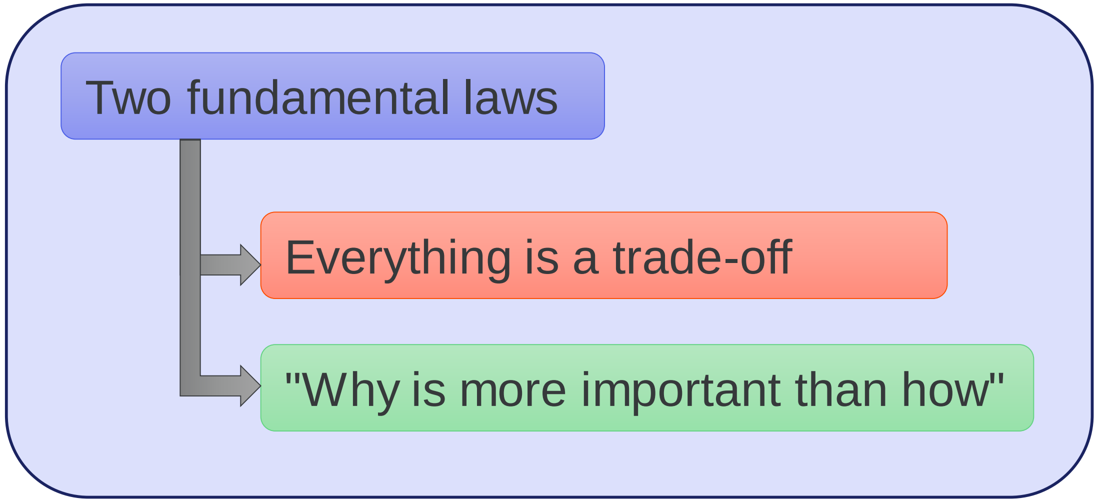


UNDERSTANDING SOFTWARE ARCHITECTURE

WHAT'S SOFTWARE ARCHITECTURE

Software architecture refers to the fundamental structures of a software system and the discipline of creating such structures. It encompasses the set of **significant decisions about the organization of a software system**, including the selection of structural elements and their interfaces, as well as their behavior and collaboration

Software architecture is about making fundamental **structural choices that are costly to change once implemented**.



Two fundamental laws

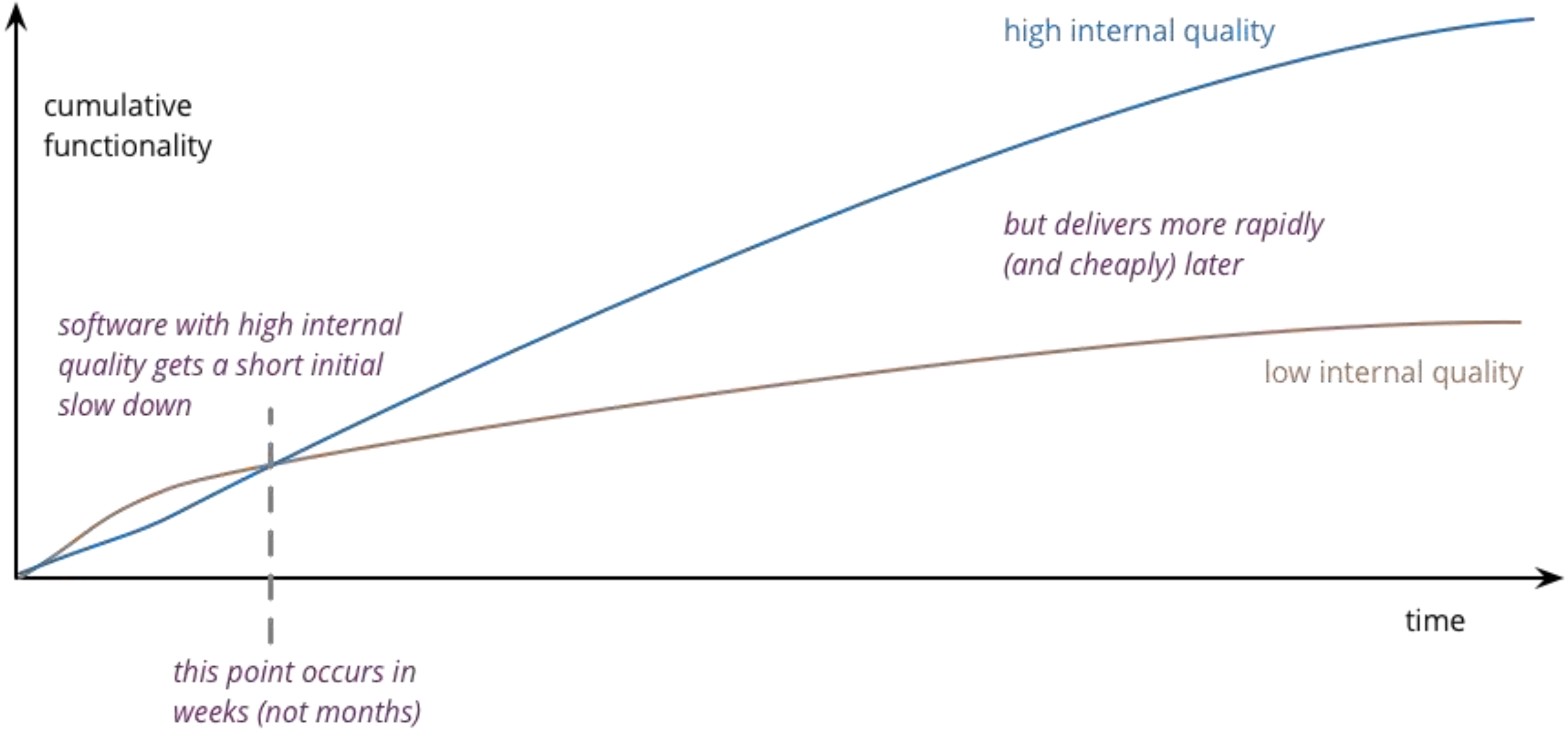
Everything is a trade-off

"Why is more important than how"

en.wikipedia.org

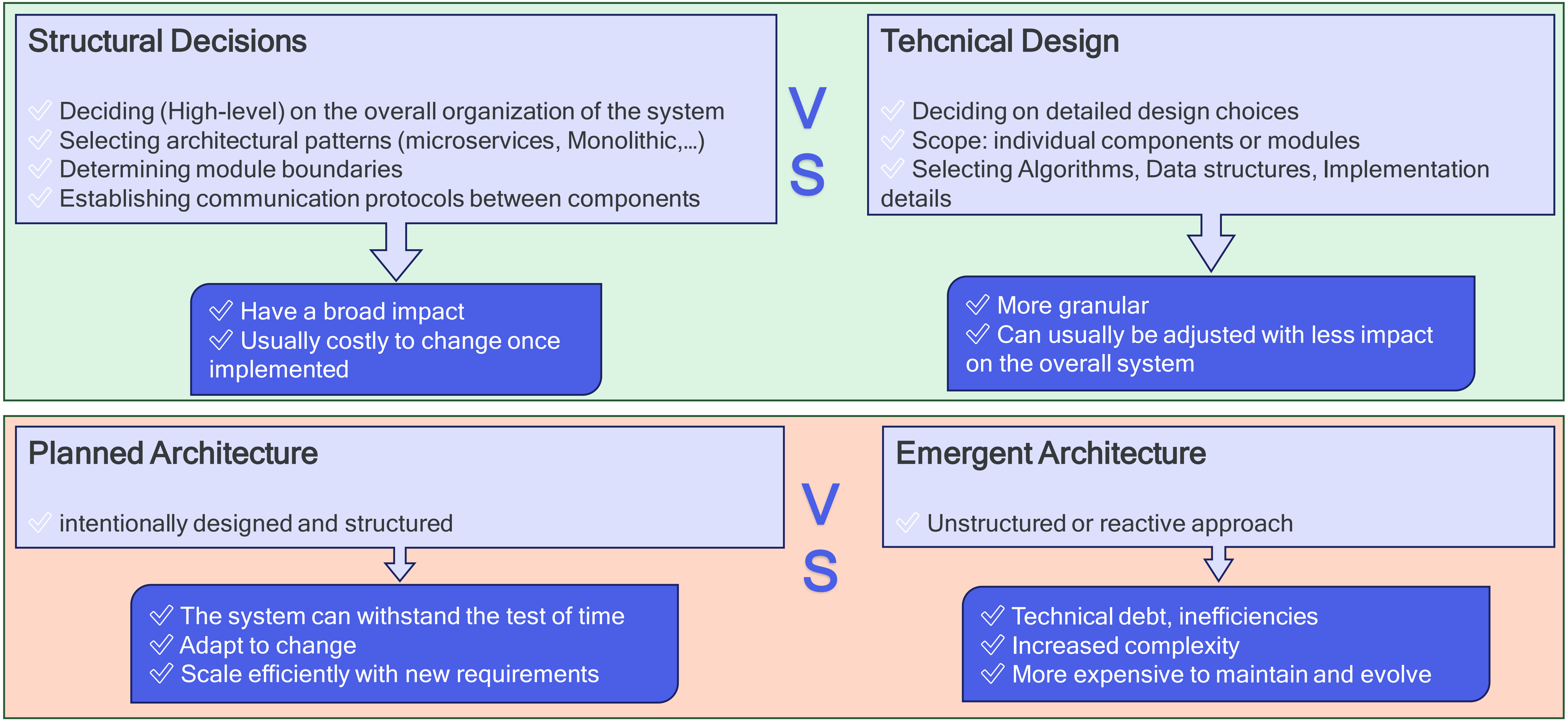
WHY DOES IT MATTER?

- ✓ Foundation for System Quality
- ✓ Facilitates Communication
- ✓ Guides Development



martinfowler.com

WHY DOES IT MATTER?



Structural Decisions

- ✓ Deciding (High-level) on the overall organization of the system
- ✓ Selecting architectural patterns (microservices, Monolithic,…)
- ✓ Determining module boundaries
- ✓ Establishing communication protocols between components

↓

- ✓ Have a broad impact
- ✓ Usually costly to change once implemented

VS

Tehcnical Design

- ✓ Deciding on detailed design choices
- ✓ Scope: individual components or modules
- ✓ Selecting Algorithms, Data structures, Implementation details

↓

- ✓ More granular
- ✓ Can usually be adjusted with less impact on the overall system

Planned Architecture

- ✓ intentionally designed and structured

↓

- ✓ The system can withstand the test of time
- ✓ Adapt to change
- ✓ Scale efficiently with new requirements

VS

Emergent Architecture

- ✓ Unstructured or reactive approach

↓

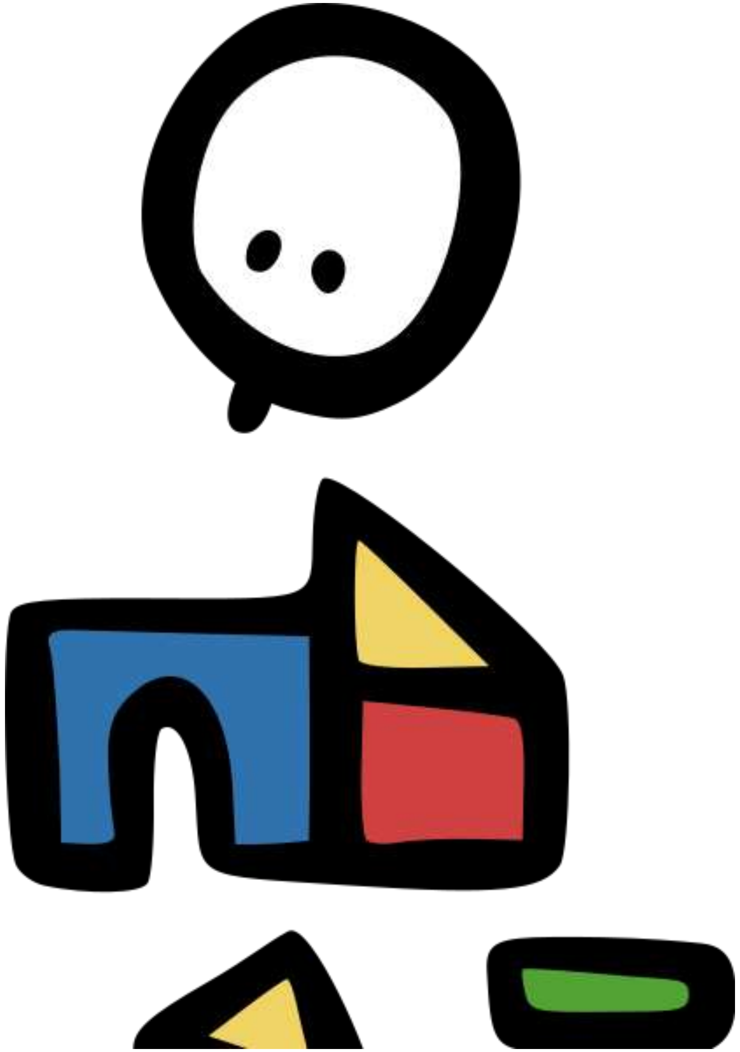
- ✓ Technical debt, inefficiencies
- ✓ Increased complexity
- ✓ More expensive to maintain and evolve

Fundamentals of Software Architecture on geeksforgeeks.org

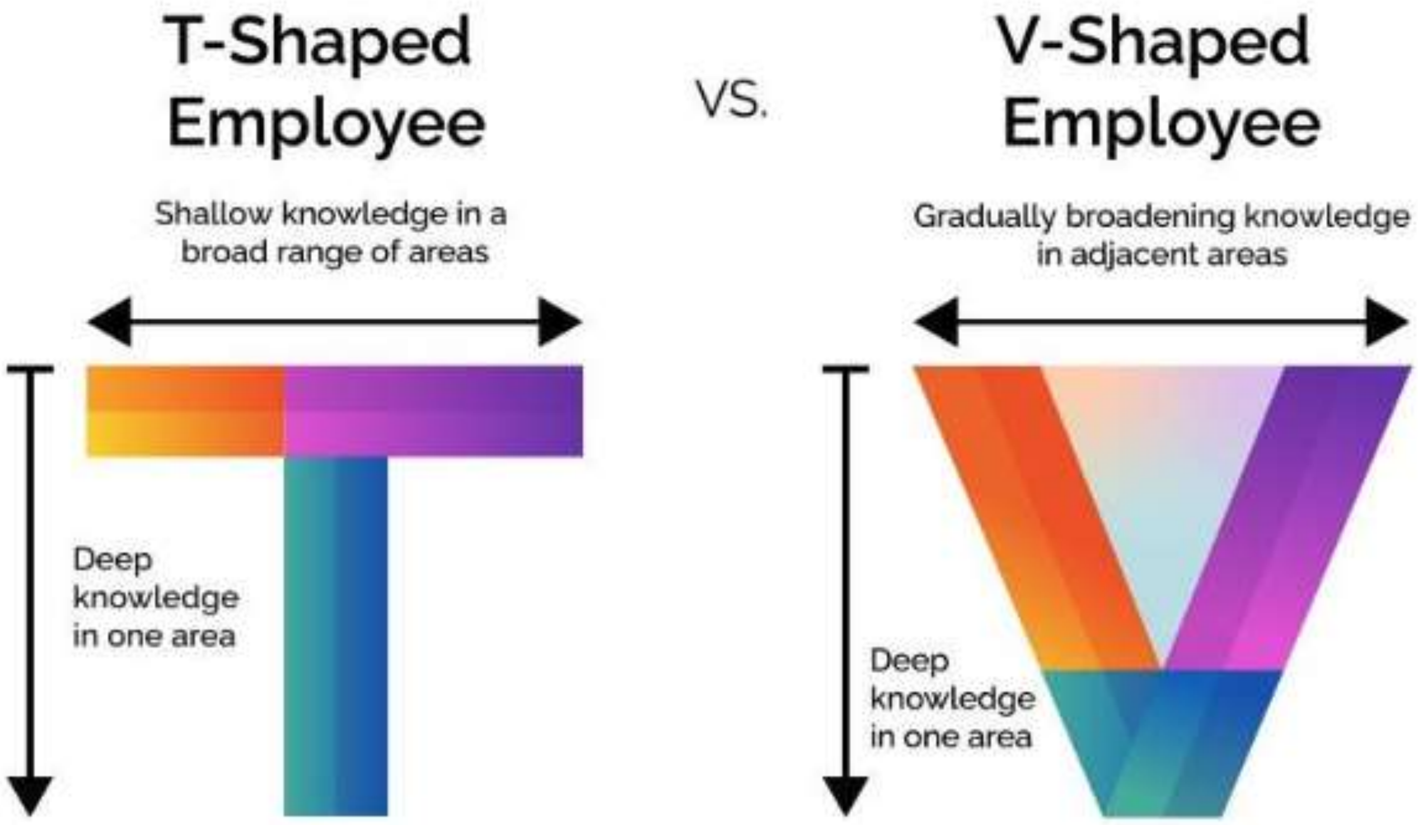
THE SOFTWARE ARCHITECT ROLE

Is the person (or AI ☺) responsible for making high-level **significant** decisions that affect:

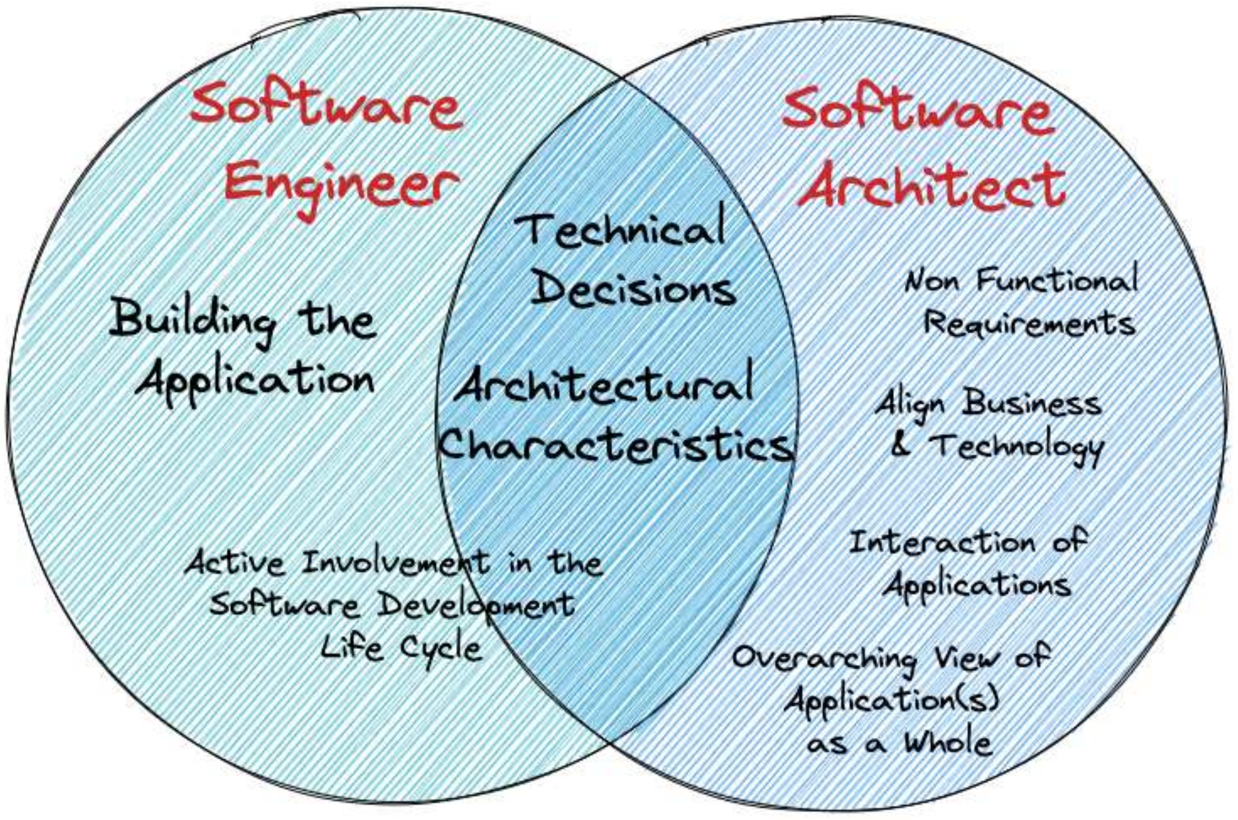
- Quality
- Security
- Performance
- **Maintainability & extensibility**



SOFTWARE ARCHITECT PROFILE


T-Shaped Employee
VS.
V-Shaped Employee
Shallow knowledge in a broad range of areas
Deep knowledge in one area
Gradually broadening knowledge in adjacent areas
Deep knowledge in one area

ARCHITECT OR SENIOR DEV?


Software Engineer
Building the Application
Active Involvement in the Software Development Life Cycle
Technical Decisions
Architectural Characteristics
Software Architect
Non Functional Requirements
Align Business & Technology
Interaction of Applications
Overarching View of Application(s) as a Whole

(c) CloudWayDigital.com

> ARCHITECTURE DEALS WITH THE **HUMAN** AND ORGANIZATIONAL SIDE OF SOFTWARE DEVELOPMENT: **TALKING TO PEOPLE** ABOUT THE PROBLEMS THEY WANT SOLVED AND DESIGNING A SOLUTION TO THOSE PROBLEMS

https://www.oreilly.com/radar/software-architecture-in-an-ai-world/

ARCHITECTURE AT DIFFERENT LEVELS

- Enterprise Architecture
- System Architecture
- Component-Level Architecture
- Others - Data Architect, Cloud Architect, Solution Architect

ENTERPRISE ARCHITECTURE

GOALS

Strategy Alignment: Ensuring that IT initiatives support business goals.

TOOLS

- **Input**: Architecture trend papers. Ex: Gartner reports
- **Outputs**: C4 models, ADR, Wardley maps

EXAMPLES

- Cloud provider partnership decision

SYSTEM ARCHITECTURE

GOALS

- Designing and optimizing the technical **structure** of a system, ensuring it meets performance, security & regulation, and scalability requirements

TOOLS

- **Inputs:** Papers and ADRs
- **Outputs**: ADRs, arch diagrams – Plantuml

EXAMPLES

- Microservices vs. Monoliths architecture

COMPONENT ARCHITECTURE

GOALS

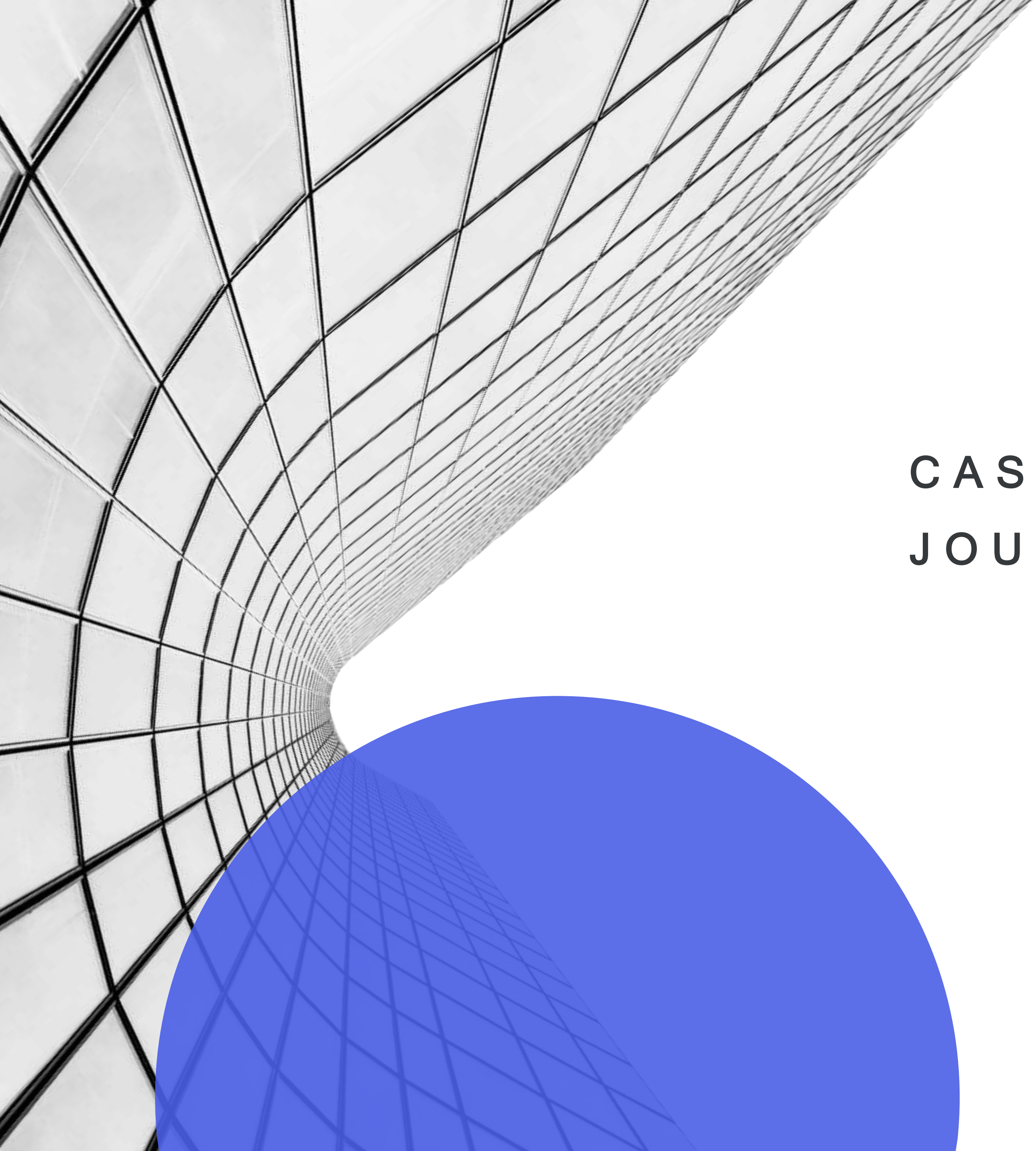
- Deals with the design and interaction of individual software components within a system. It emphasizes modularity, reusability, and maintainability

TOOLS

- **Inputs**: papers, trend reports, tech radars
- **Outputs**: Component tech documentation

EXAMPLES

- Microfrontend architecture



CASE STUDY: THE ASSURE JOURNEY

INSURANCE SOFTWARE & BPS

#1 Insurance IT provider (40+y)

1300+ customers

100M insurance policies **19M+** policies (only in AWS)

70+ countries

TIMELINE

1980 2015	Delivering PAS under license + Professional services
2015	Digital First - Api enablement + Omnichannel
2018	AWS Partnership - Cloudification
2020	Move into SaaS + IDP
2025	PaaS ?

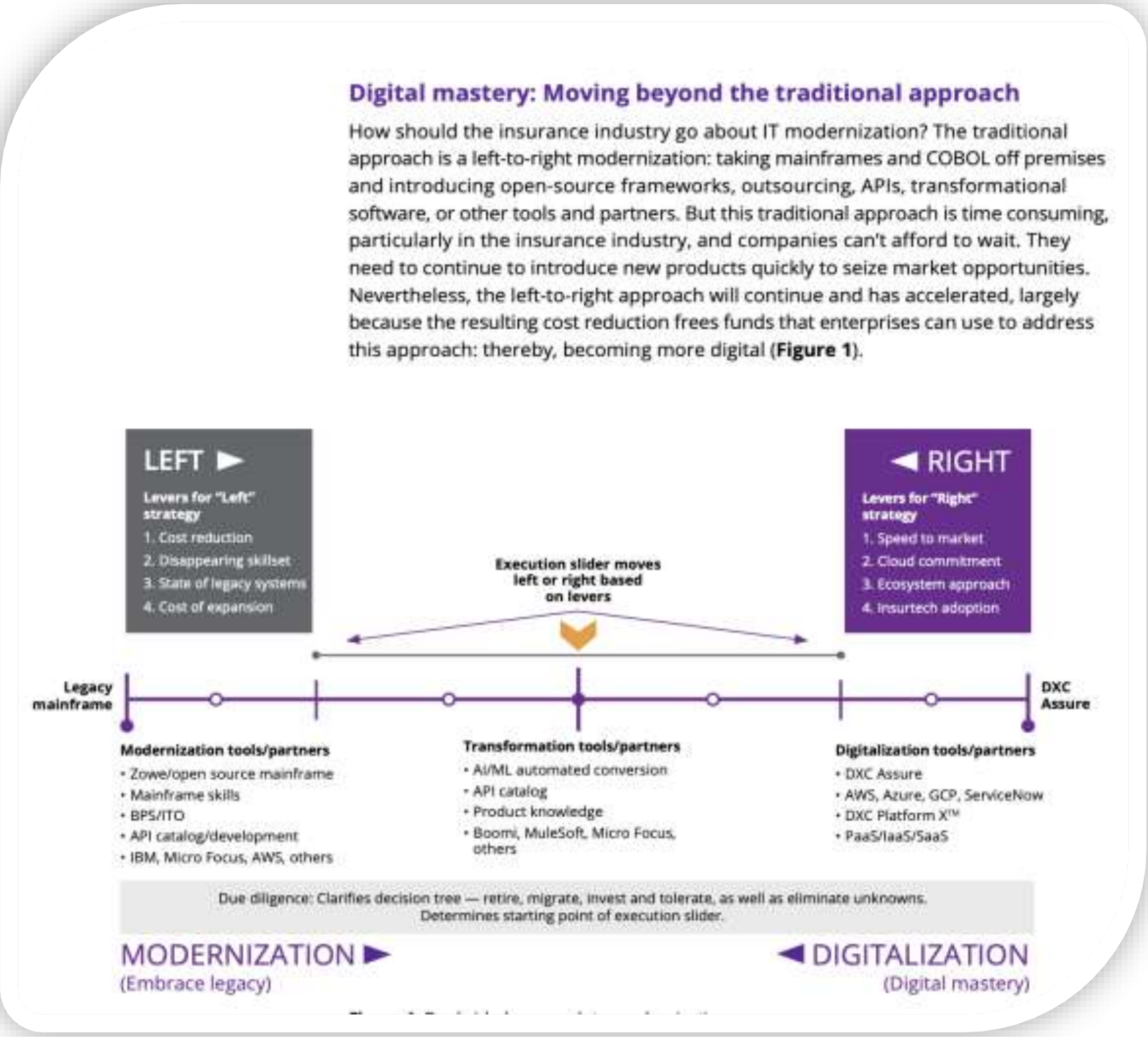
THE INSURANCE SOFTWARE MODERNIZATION STRATEGY

3 PRIORITIES

Modernization of Legacy Systems

Adoption of Cloud Computing and aaS Models

AI-driven implementation



Digital mastery: Moving beyond the traditional approach

How should the insurance industry go about IT modernization? The traditional approach is a left-to-right modernization: taking mainframes and COBOL off premises and introducing open-source frameworks, outsourcing, APIs, transformational software, or other tools and partners. But this traditional approach is time consuming, particularly in the insurance industry, and companies can't afford to wait. They need to continue to introduce new products quickly to seize market opportunities. Nevertheless, the left-to-right approach will continue and has accelerated, largely because the resulting cost reduction frees funds that enterprises can use to address this approach: thereby, becoming more digital (**Figure 1**).

LEFT ►

Levers for "Left" strategy

1. Cost reduction
2. Disappearing skillset
3. State of legacy systems
4. Cost of expansion

◄ RIGHT

Levers for "Right" strategy

1. Speed to market
2. Cloud commitment
3. Ecosystem approach
4. Insurtech adoption

Execution slider moves left or right based on levers

Legacy mainframe

DXC Assure

Modernization tools/partners

- Zowe/open source mainframe
- Mainframe skills
- BPS/ITO
- API catalog/development
- IBM, Micro Focus, AWS, others

Transformation tools/partners

- AI/ML automated conversion
- API catalog
- Product knowledge
- Boomi, MuleSoft, Micro Focus, others

Digitalization tools/partners

- DXC Assure
- AWS, Azure, GCP, ServiceNow
- DXC Platform X™
- PaaS/IaaS/SaaS

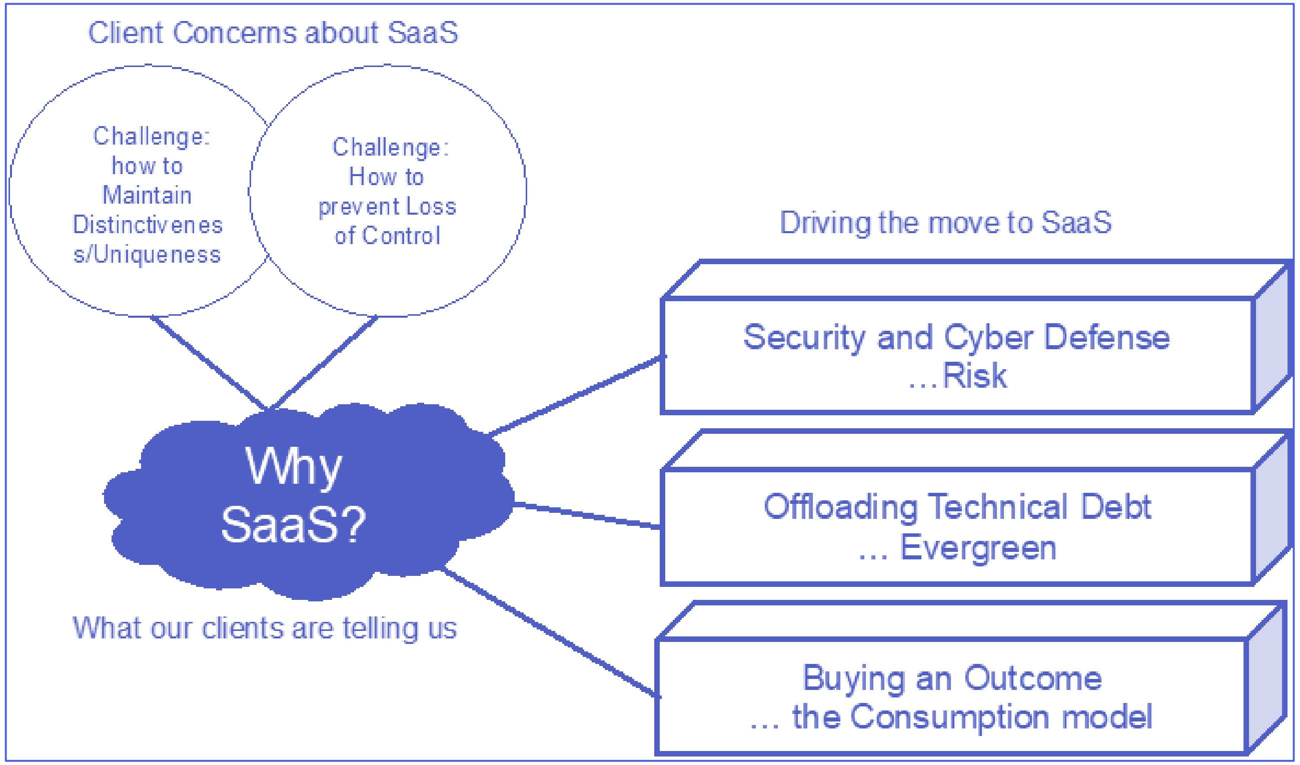
Due diligence: Clarifies decision tree — retire, migrate, invest and tolerate, as well as eliminate unknowns.
Determines starting point of execution slider.

MODERNIZATION ►
(Embrace legacy)

◄ DIGITALIZATION
(Digital mastery)

https://dxc.com/content/dam/dxc/projects/dxc-com/us/pdfs/industries/insurance/A-winning-strategy-for-modernizing-insurance-IT.pdf

CHALLENGES



Client Concerns about SaaS
Challenge: how to Maintain Distinctiveness/Uniqueness
Challenge: How to prevent Loss of Control
Why SaaS?
What our clients are telling us
Driving the move to SaaS
Security and Cyber Defense …Risk
Offloading Technical Debt … Evergreen
Buying an Outcome … the Consumption model

Heterogeneous architecture

Multiple technology stacks

+80 product portfolio

+20 global distributed product teams

Staffing teams for an IDP - Assure Platform

- architecture teams to design it
- Engineering teams to build it
- SaasOps teams to operate it for real clients

THE STRATEGIC ARCHITECTURAL DECISIONS

P-E-T Model: DXC offers a global suite of insurance software products designed to manage the entire insurance lifecycle. DXC is committed in supporting insurers in **protecting, extending, and transforming** their application landscapes.

How? The key enablers

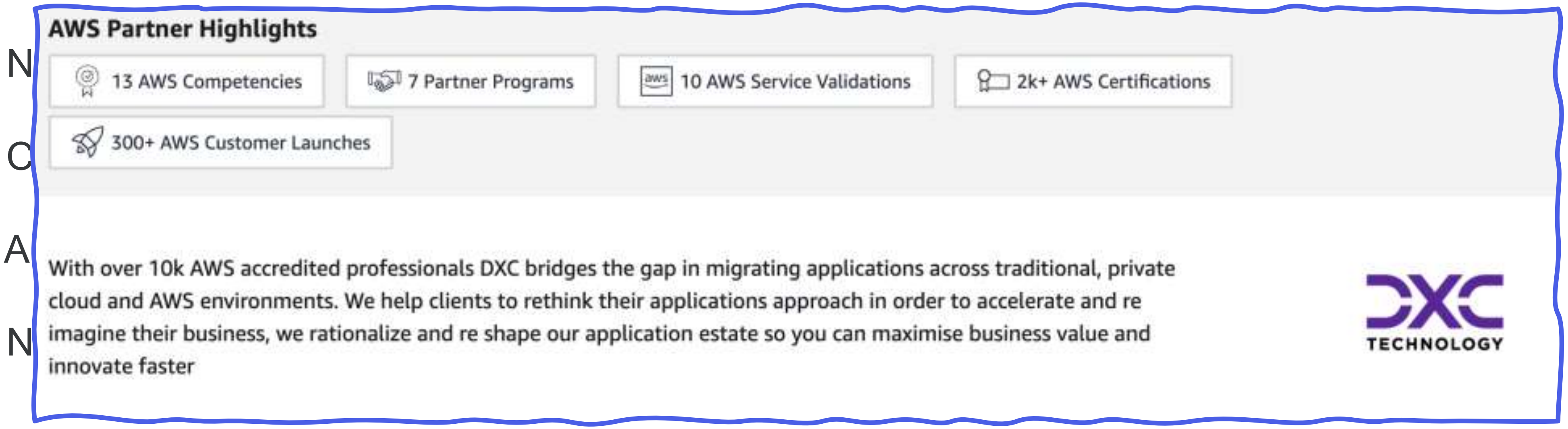
✓ Partnership with AWS as cloud provider (33% market share), at the edge of innovation

✓ N

✓ C

✓ A

✓ N



AWS Partner Highlights

- 13 AWS Competencies
- 7 Partner Programs
- 10 AWS Service Validations
- 2k+ AWS Certifications
- 300+ AWS Customer Launches

With over 10k AWS accredited professionals DXC bridges the gap in migrating applications across traditional, private cloud and AWS environments. We help clients to rethink their applications approach in order to accelerate and re imagine their business, we rationalize and re shape our application estate so you can maximise business value and innovate faster

https://aws.amazon.com/partners/dxc/

THE CORE OF THE STRATEGY

ASSURE INTERNAL DEVELOPMENT PLATFORM

The Assure Platform is a suite of well integrated but independent software products that help developers build better and faster SaaS insurance software that can be operated at scale

Assure Platform Reference Architecture

UX

- React
- HTML 5

Assure Halstack design system provides a rich library of reusable React UI components for consistent persona based UX and development accelerators.

Customer Identity Access Management

- Amazon Cognito
- AWS Amplify + React
- AWS Lambda

Assure offers customizable authentication, registration MFA, and password recovery workflows. Assure IAM can integrate with 3rd party IdPs.

Infrastructure

- Microsoft Windows
- Linux
- MICRO FOCUS Enterprise Server, CICS
- docker
- Rocket software
- AWS M2
- AWS Lambda
- Amazon EC2
- AWS Elastic Container Service

Scalable, elastic and cost optimized platform. Linux before Windows. Lambda before ECS, ECS before EC2. Each application component needs to scale out based on current or anticipated usage.

Assure 360

- Athena
- Quicksight
- AWS Glue
- Amazon DynamoDB
- Amazon CloudWatch
- AWS Lambda
- AWS Amplify + React

360 Provides a complete data centric Operational Dashboard with data visualization capabilities. The dashboard provides cloud resources utilization by products and services.

Assure Cyber Defense

- API Gateway
- AWS WAF
- GuardDuty
- Inspector
- Amazon Macie
- AWS CloudTrail
- PRISMA CLOUD
- checkov
- OWASP Open Web Application Security Project
- splunk>

Cloud security posture management (CSPM) service, secure infrastructure provisioning, logging scans, monitoring, secure standard checks and escalation. PCI compliance checks.

DevOps, Observability, Incident Escalation

- sonarqube
- HashiCorp Terraform
- JFrog ARTIFACTORY
- GitHub
- servicenow.
- docker
- Jenkins

The Assure Platform Fabric is a suite of well integrated but independent software products that help developers build better and faster SaaS insurance software that can be operated at scale.

THE ASSURE PRINCIPLES

THE THREE LAWS OF SAAS

- ⚖ **First Law**

  You must never alter the immutable product and/or solution base release (on-demand or scheduled) or its deployable AMS packaging.

  ✓ **Automation Management Services** - Set of tools and processes that support the whole SDLC process, from build to deployment.

- ⚖ **Second Law**

  Client-specific configurations and coded enhancements shall be packaged and managed within AMS without breaking law #1

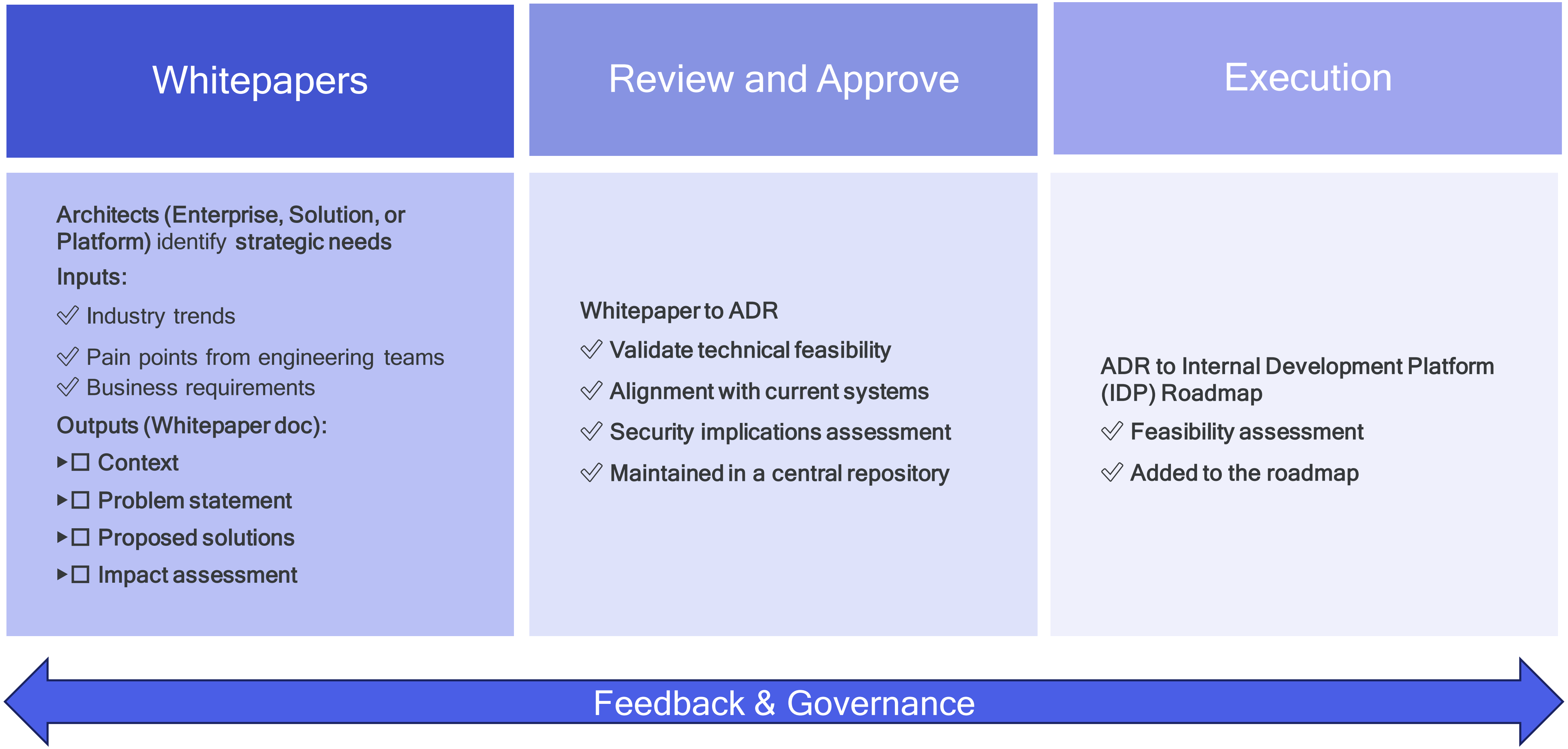
- ⚖ **Third Law**

  You must not add activities, components or packages which require manual intervention during deployment and/or are managed outside AMS, without a validated reason, and if permitted must never break the first 2 Laws.

- ⚖ **Corollary to the 3 Laws**

  Releases must always be backward compatible to ensure low-risk upgrades without disruption, manual intervention or breaking the client code, unless with a validated reason, and if permitted must never break the first 2 Laws.

THE ASSURE PRINCIPLES
THE DECISION-MAKING PROCESS



Whitepapers

Architects (Enterprise, Solution, or Platform) identify **strategic needs**

Inputs:

- ✓ Industry trends
- ✓ Pain points from engineering teams
- ✓ Business requirements

Outputs (Whitepaper doc):

- ▶ ☐ Context
- ▶ ☐ Problem statement
- ▶ ☐ Proposed solutions
- ▶ ☐ Impact assessment

Review and Approve

Whitepaper to ADR

- ✓ Validate technical feasibility
- ✓ Alignment with current systems
- ✓ Security implications assessment
- ✓ Maintained in a central repository

Execution

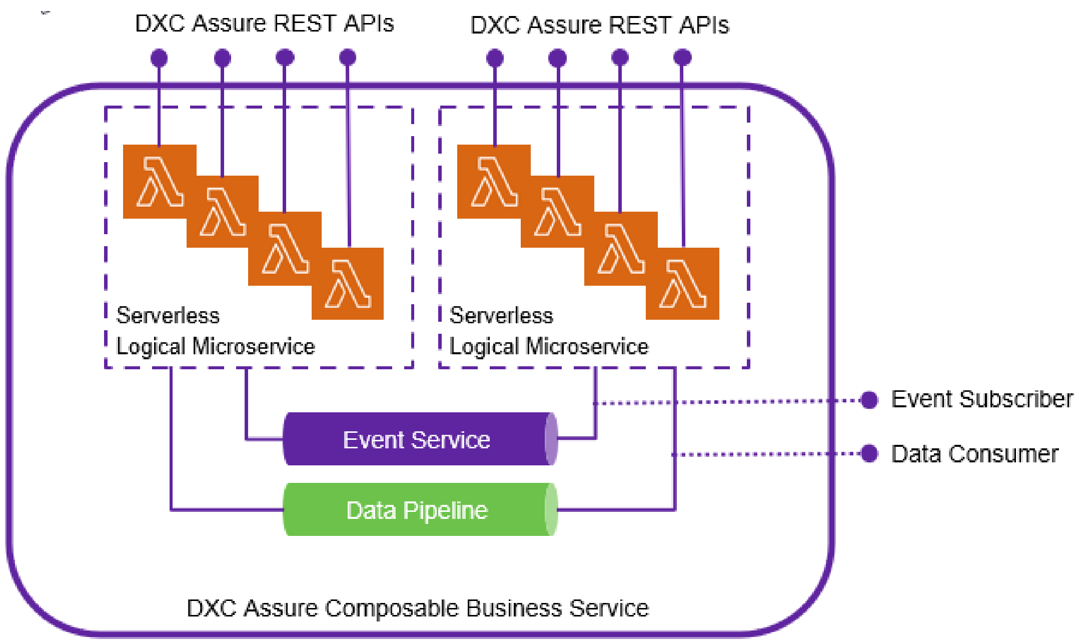
ADR to Internal Development Platform (IDP) Roadmap

- ✓ Feasibility assessment
- ✓ Added to the roadmap

Feedback & Governance

THE ASSURE PRINCIPLES

CLOUD FIRST & SAAS



DXC Assure SaaS Optimized Target Architecture Model

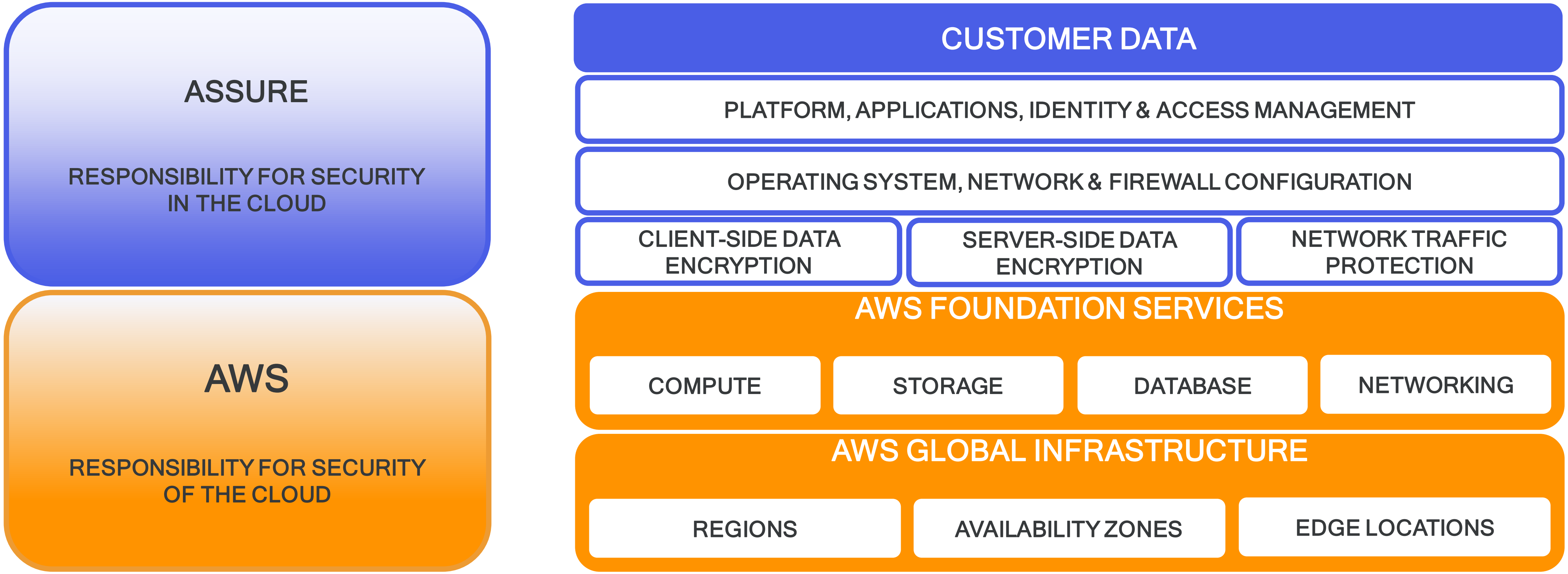
Key pillars

- Multi-Tenancy & Isolation Strategy 🏢🔒
- Scalability & Elasticity 📈⚖️▯
- SaaS Identity & Access Management 🔐🆔
- Observability & Tenant-Level Monitoring 🔍📊
- Resilience & High Availability ⚡💾
- AWS Well-Architected Framework Alignment ✅▯
- CI/CD & Automated Deployments 🚀⚙️▯
- Cost Efficiency & Billing Transparency 💰📉
- Data Security & Compliance 🔒📜
- AI & Future-Proofing ▯🔮

THE ASSURE PRINCIPLES

SECURE.- SHARED RESPONSIBILITY MODEL

Security is the **foundation of all SaaS design decisions**, ensuring that every layer of the system–identity, data, infrastructure, and operations–adheres to the **highest security standards from inception to deployment**



ASSURE

RESPONSIBILITY FOR SECURITY IN THE CLOUD

AWS

RESPONSIBILITY FOR SECURITY OF THE CLOUD

CUSTOMER DATA

PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT

OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION

CLIENT-SIDE DATA ENCRYPTION

SERVER-SIDE DATA ENCRYPTION

NETWORK TRAFFIC PROTECTION

AWS FOUNDATION SERVICES

COMPUTE

STORAGE

DATABASE

NETWORKING

AWS GLOBAL INFRASTRUCTURE

REGIONS

AVAILABILITY ZONES

EDGE LOCATIONS

https://aws.amazon.com/compliance/shared-responsibility-model/

Image: NIST CSF

THE ASSURE PRINCIPLES

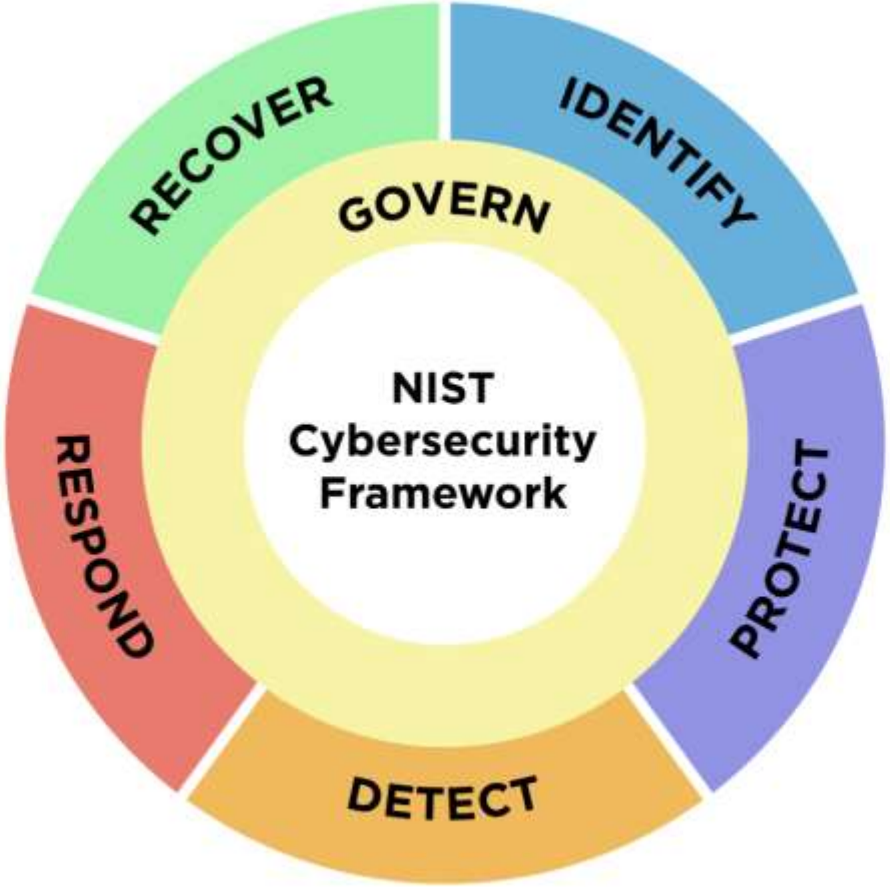
SECURE.- CYBERSECURITY ARCHITECTURE

Secure Architecture – Secure By Design with security best practices, guidelines and standards

Secure Monitoring – Security Information & Event Management (SIEM) with continuous monitoring for malicious and unauthorized activity

Secure Response – Security Incident Response (SIR) to quickly resolve and contain cybersecurity incidents

Secure Recovery – Incident Response to restore assets and operations affected by cybersecurity incidents



Secure Identity – Identity & Access Management (IAM) with authentication and access control

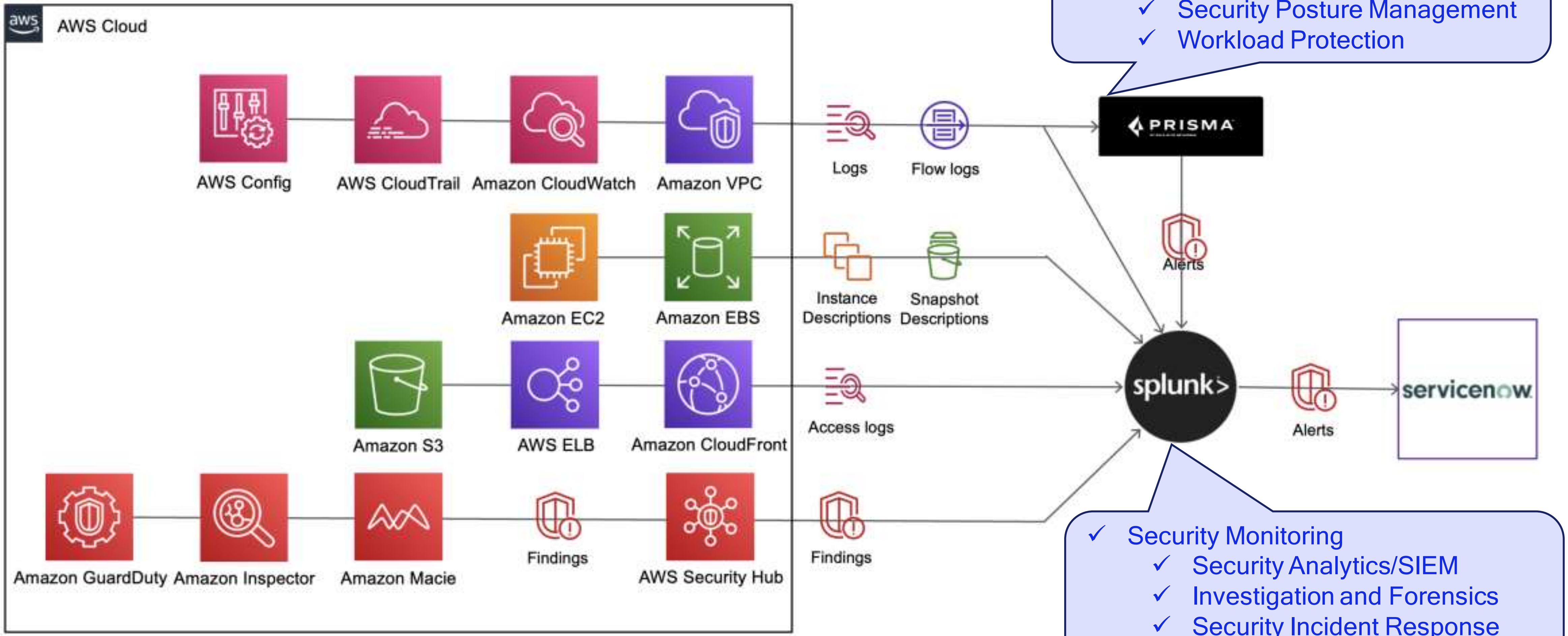
Secure Data – Data Encryption to protect data at rest and in transit

Secure Code – Vulnerability Management with automated OSS vulnerability and license compliance scans

Secure Infrastructure – Compliance Management with automated compliance scans

THE ASSURE PRINCIPLES

SECURE.- CYBER DEFENSE SOLUTION

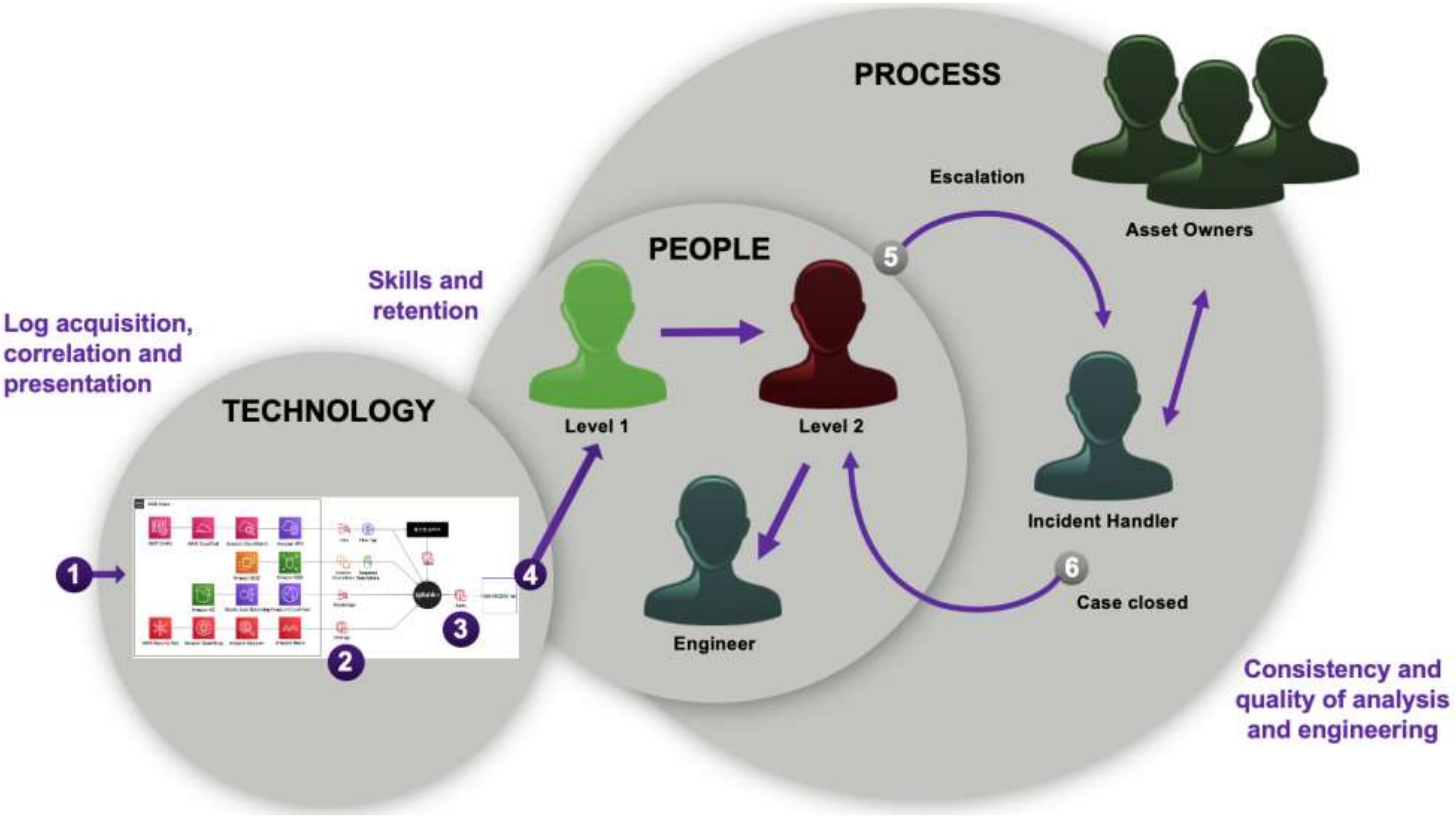


- ✓ Application Protection
  - ✓ Code Security
  - ✓ Security Posture Management
  - ✓ Workload Protection

- ✓ Security Monitoring
  - ✓ Security Analytics/SIEM
  - ✓ Investigation and Forensics
  - ✓ Security Incident Response
  - ✓ Security Operations

THE ASSURE PRINCIPLES

SECURE.- CYBER DEFENSE STRATEGY



Log acquisition, correlation and presentation

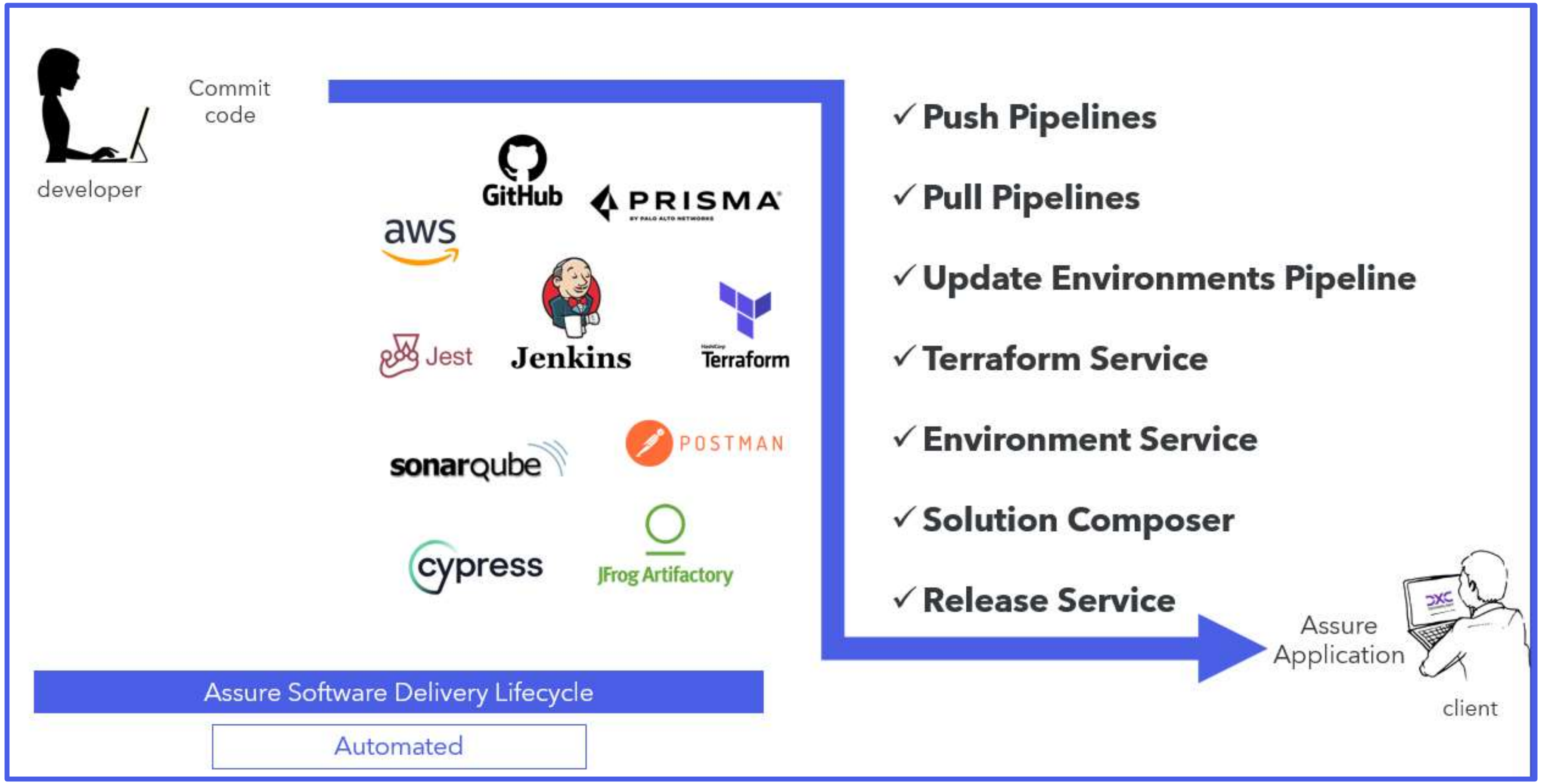
Skills and retention

Consistency and quality of analysis and engineering

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.

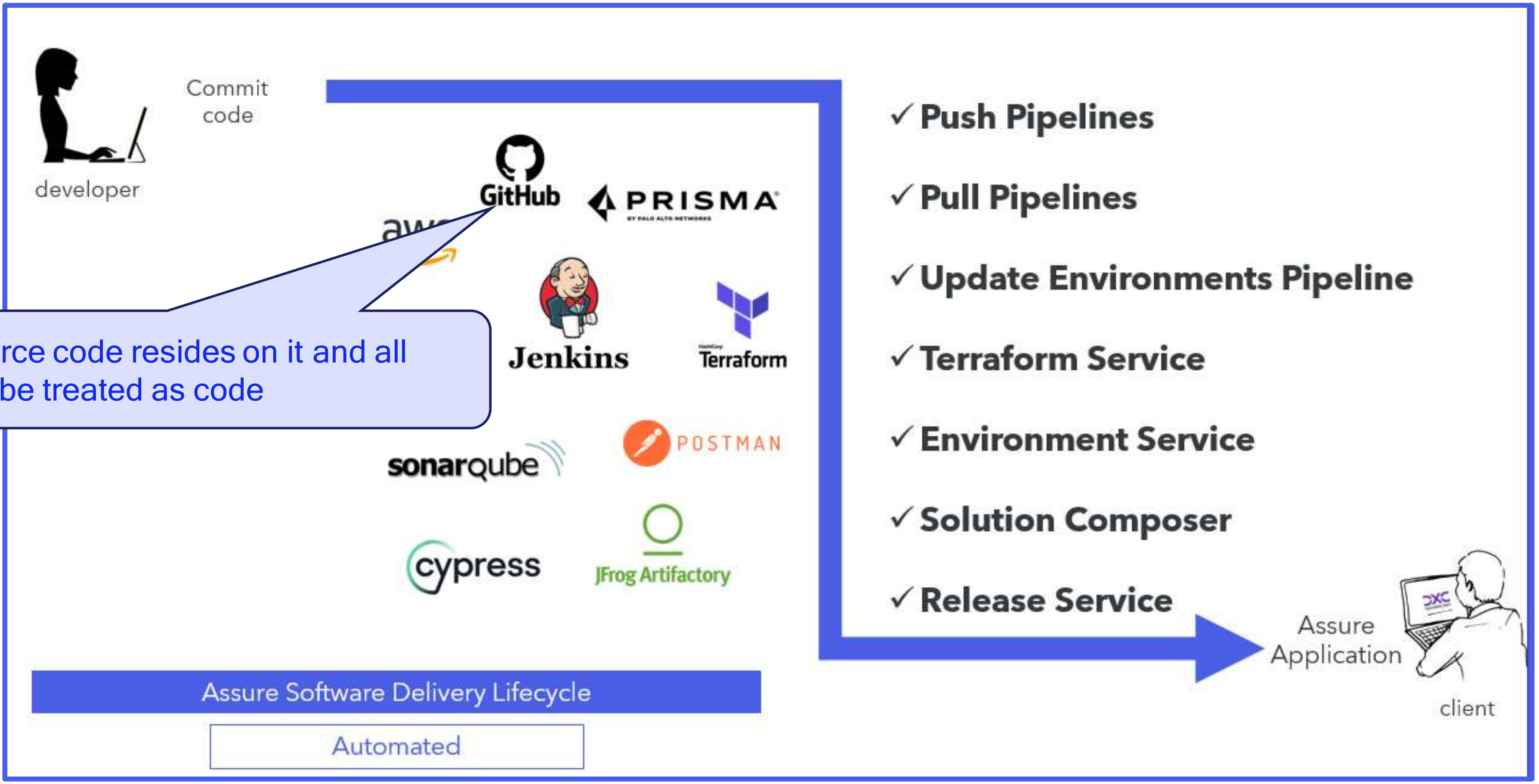


- ✓ Push Pipelines
- ✓ Pull Pipelines
- ✓ Update Environments Pipeline
- ✓ Terraform Service
- ✓ Environment Service
- ✓ Solution Composer
- ✓ Release Service

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.



- ✓ Push Pipelines
- ✓ Pull Pipelines
- ✓ Update Environments Pipeline
- ✓ Terraform Service
- ✓ Environment Service
- ✓ Solution Composer
- ✓ Release Service

Assure Software Delivery Lifecycle

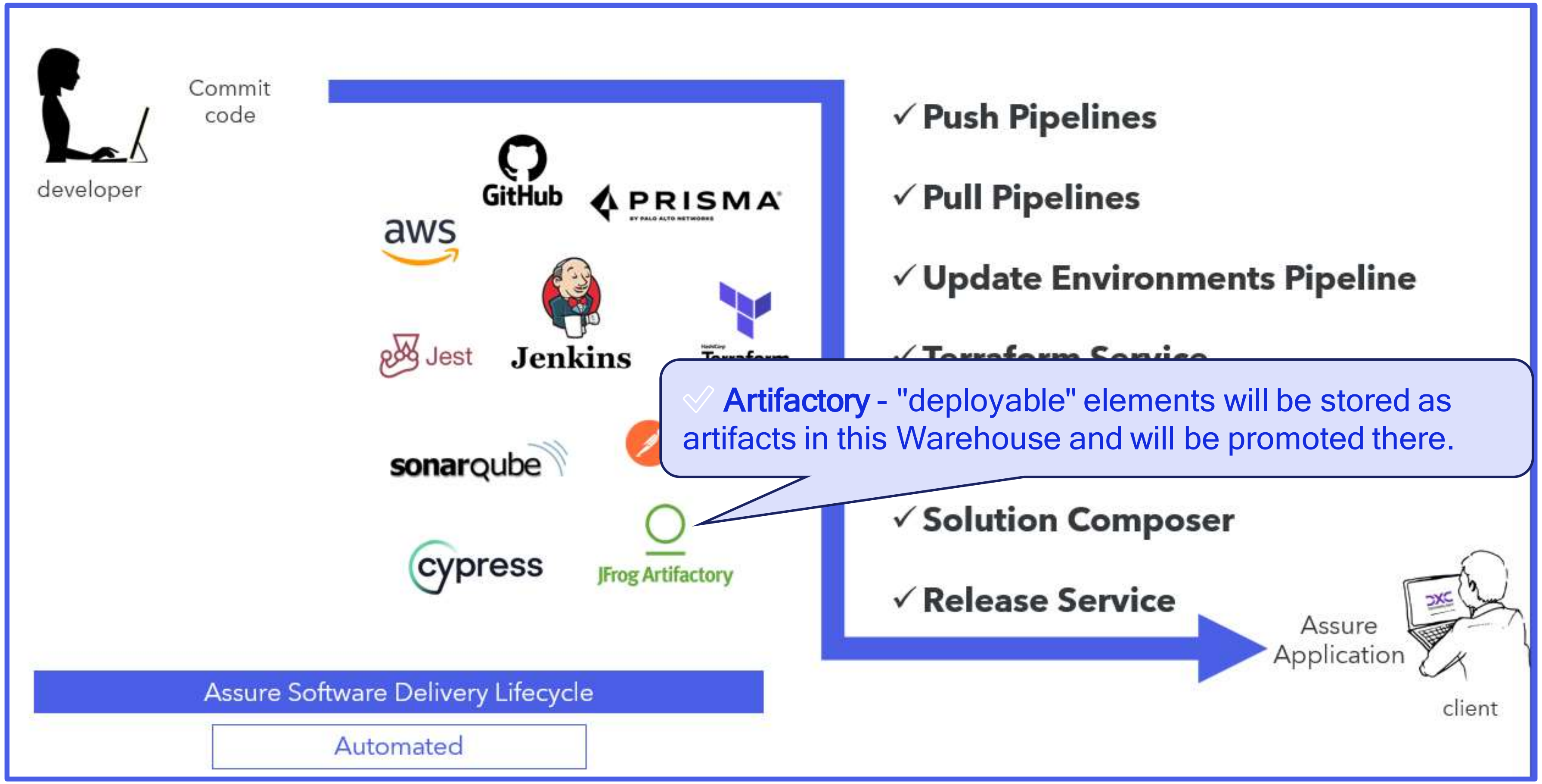
Automated

✓ GitHub - all source code resides on it and all components must be treated as code

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.



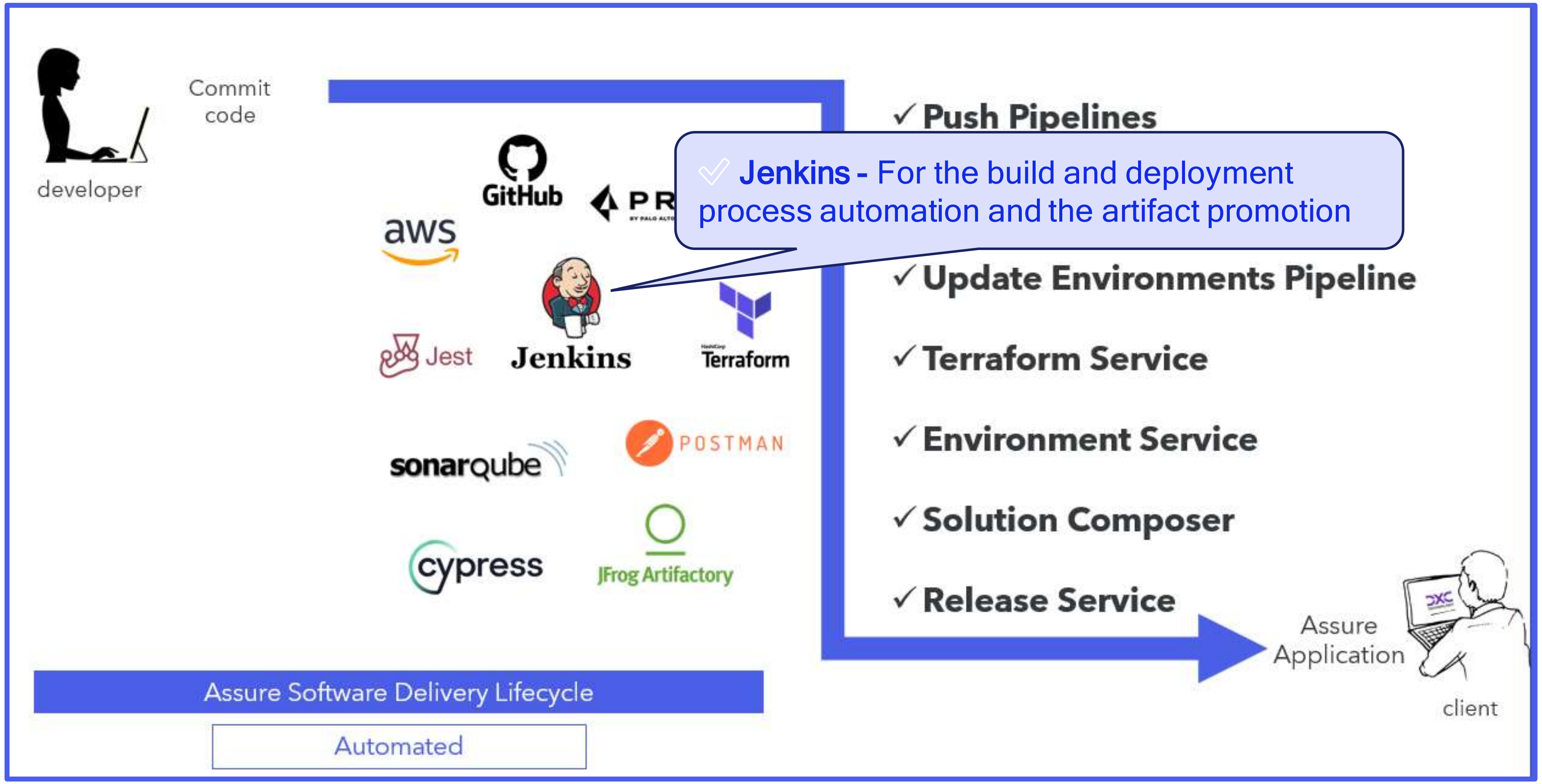
- ✓ Push Pipelines
- ✓ Pull Pipelines
- ✓ Update Environments Pipeline
- ✓ Terraform Service
- ✓ Solution Composer
- ✓ Release Service

✓ Artifactory - "deployable" elements will be stored as artifacts in this Warehouse and will be promoted there.

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.



✓ Push Pipelines

✓ Jenkins - For the build and deployment process automation and the artifact promotion

✓ Update Environments Pipeline

✓ Terraform Service

✓ Environment Service

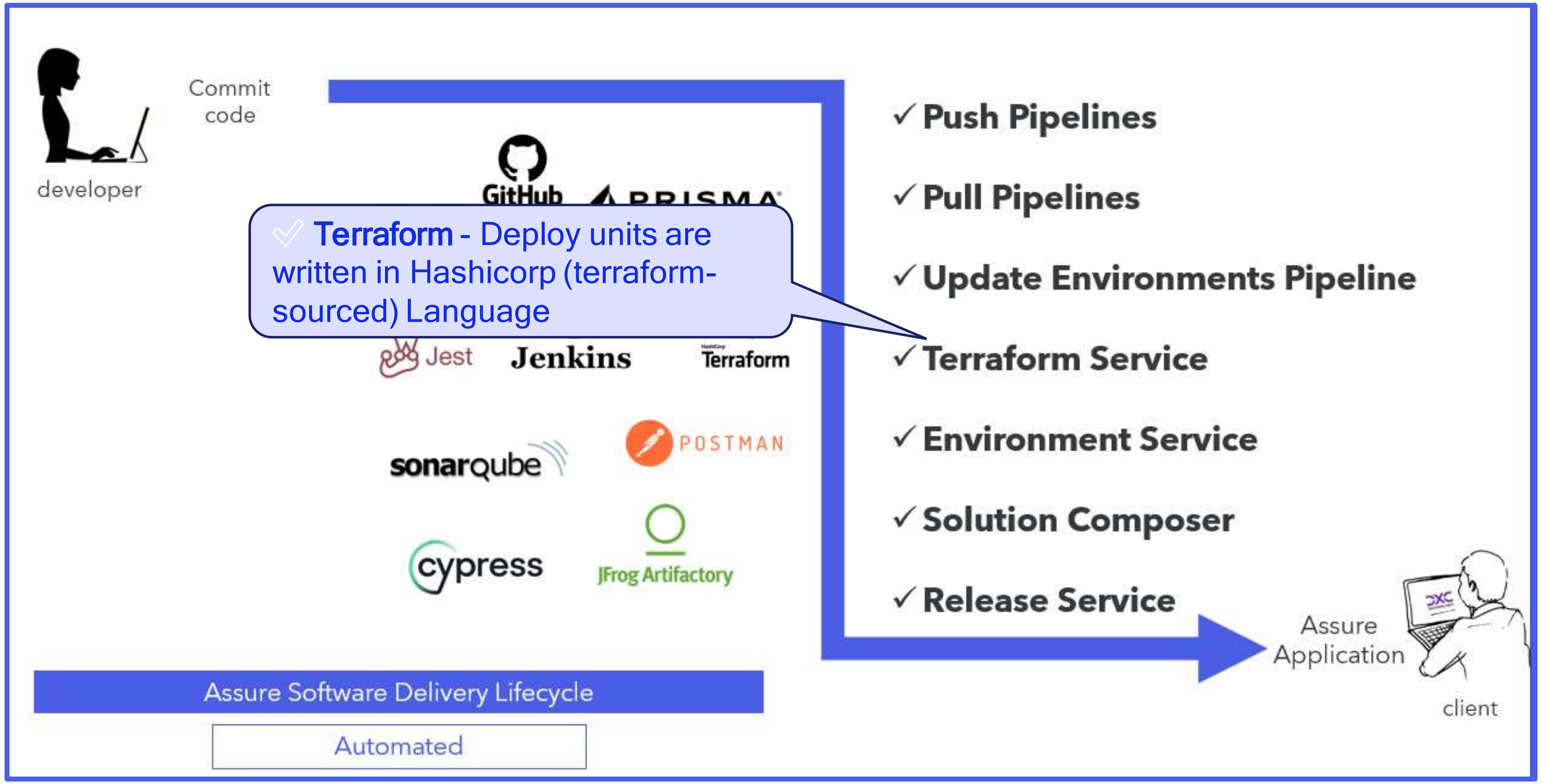
✓ Solution Composer

✓ Release Service

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.

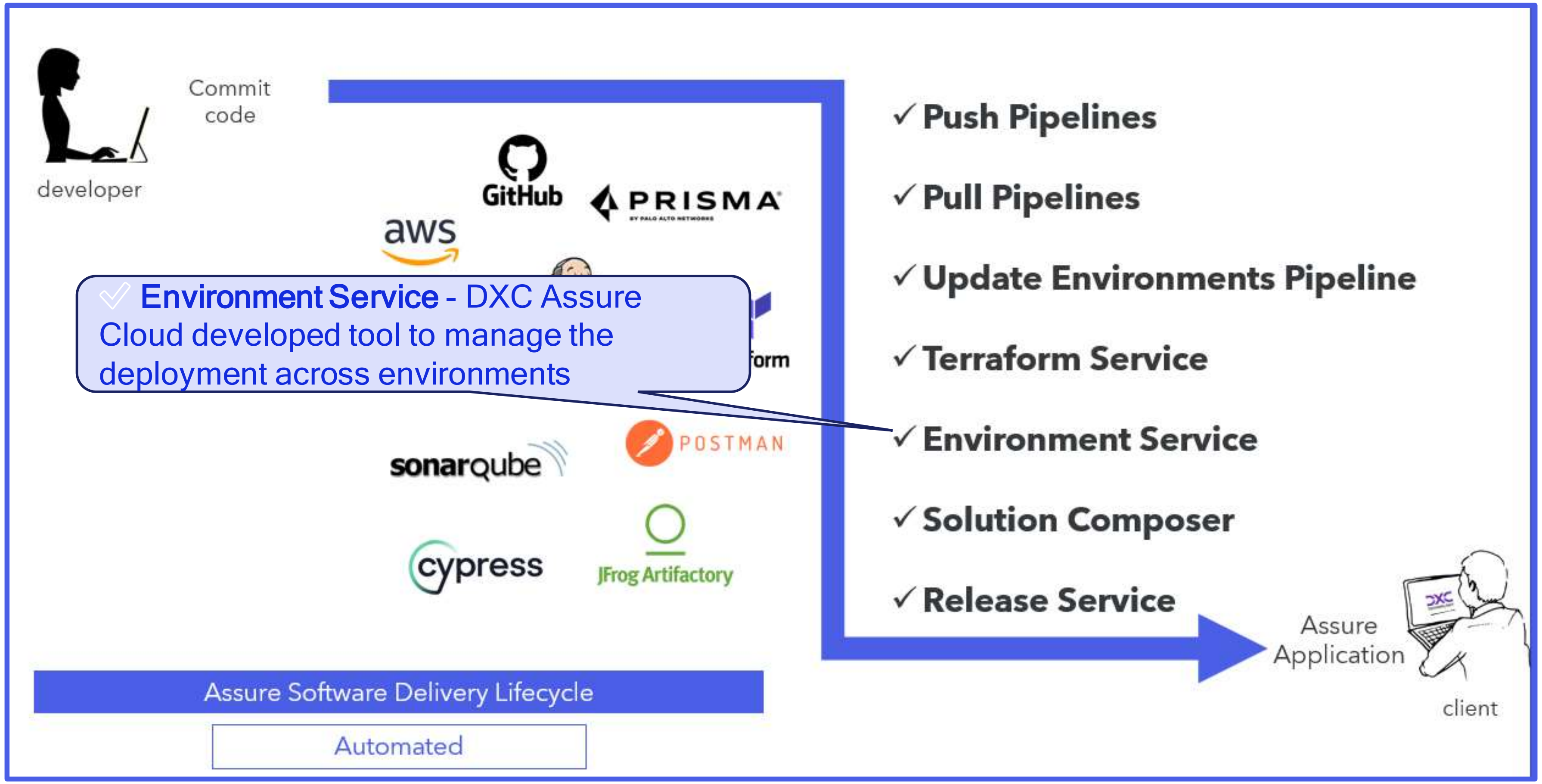


✓ Terraform - Deploy units are written in Hashicorp (terraform-sourced) Language

- ✓ Push Pipelines
- ✓ Pull Pipelines
- ✓ Update Environments Pipeline
- ✓ Terraform Service
- ✓ Environment Service
- ✓ Solution Composer
- ✓ Release Service

THE ASSURE PRINCIPLES
FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.

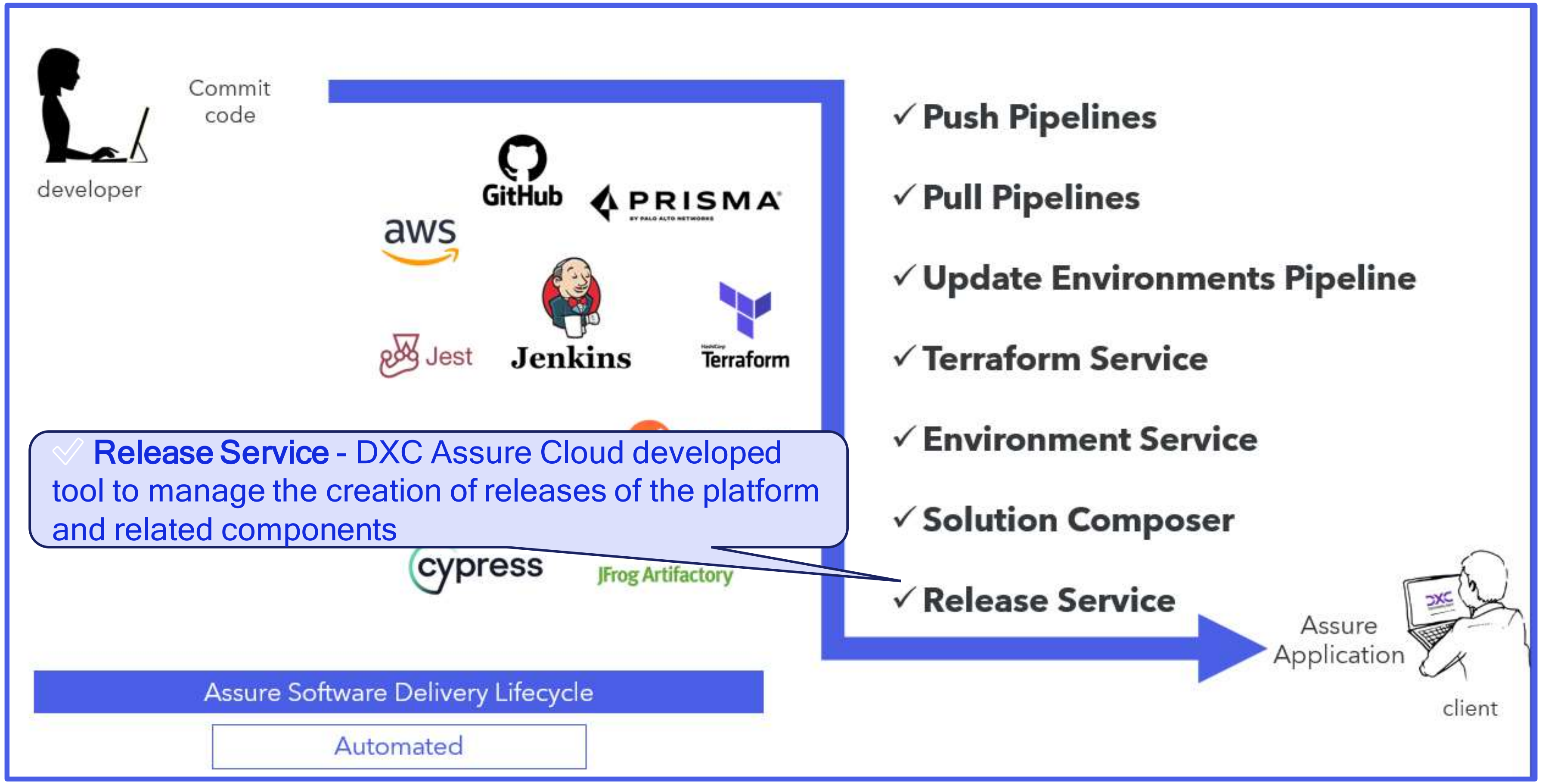


✓ Environment Service - DXC Assure Cloud developed tool to manage the deployment across environments

THE ASSURE PRINCIPLES

FULLY AUTOMATED

Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.



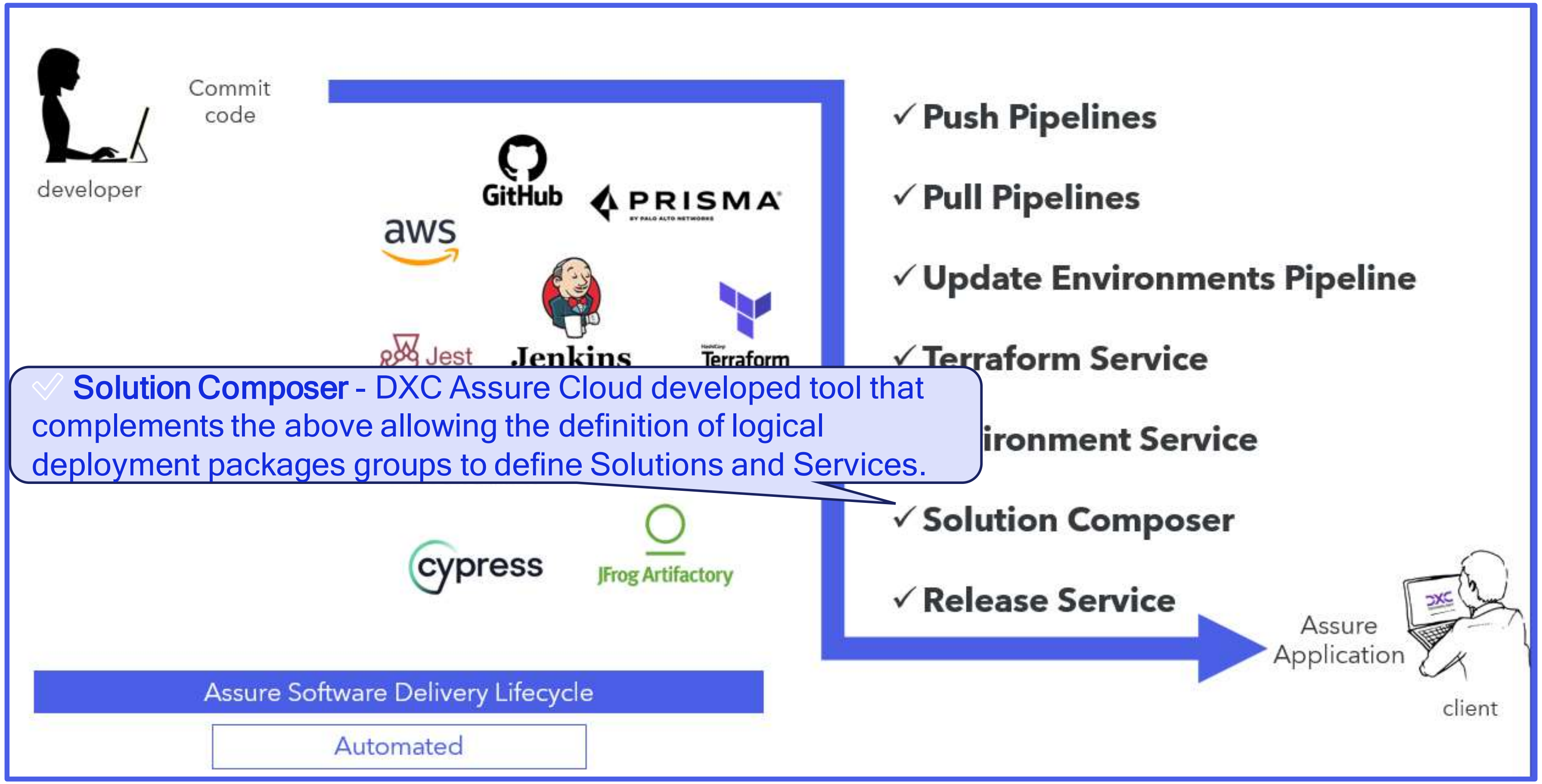
- ✓ Push Pipelines
- ✓ Pull Pipelines
- ✓ Update Environments Pipeline
- ✓ Terraform Service
- ✓ Environment Service
- ✓ Solution Composer
- ✓ Release Service

✓ **Release Service** - DXC Assure Cloud developed tool to manage the creation of releases of the platform and related components

THE ASSURE PRINCIPLES

FULLY AUTOMATED

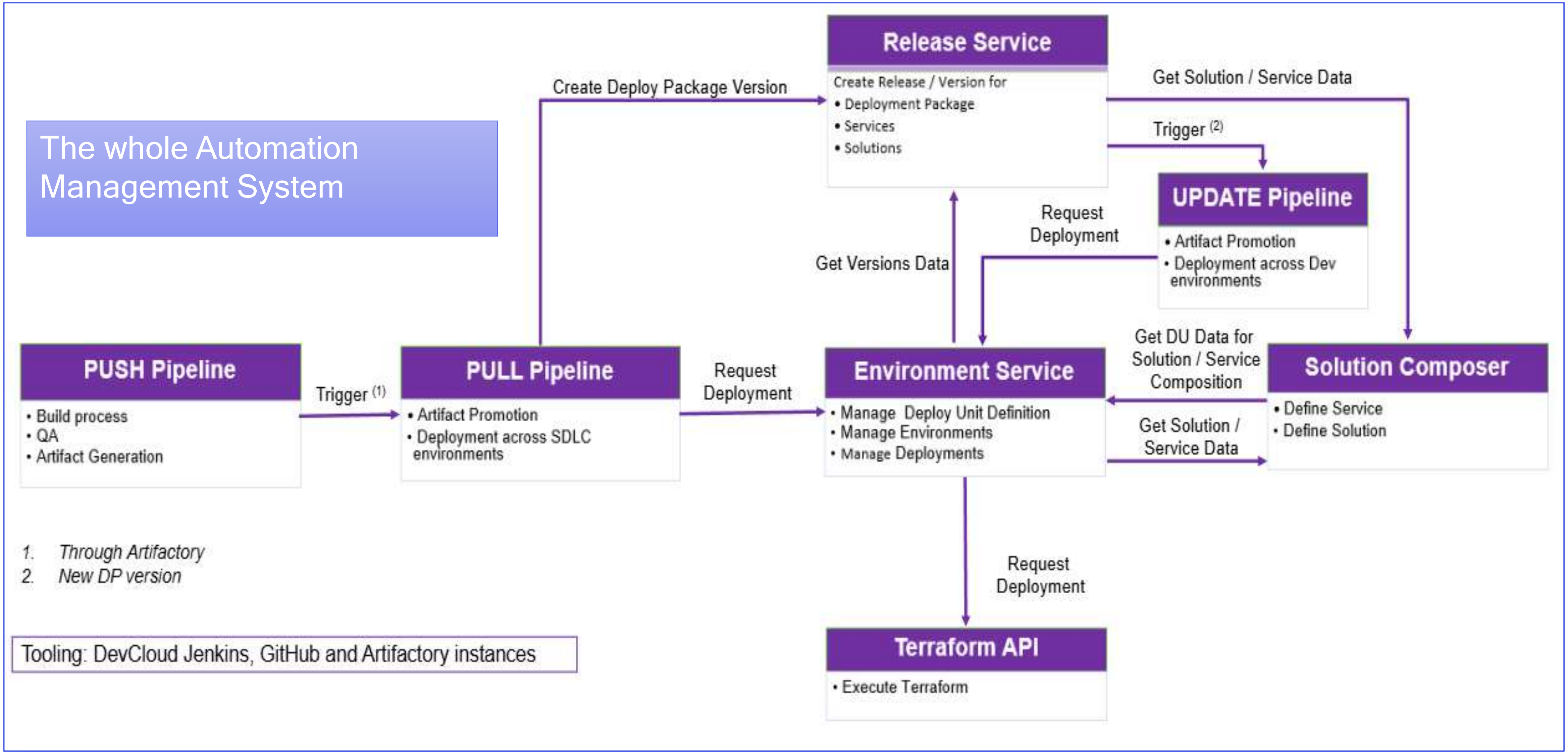
Goal: complete automation of the build, release and deploy processes required to go from requirements and code to a working application in customer environments.



✓ Solution Composer - DXC Assure Cloud developed tool that complements the above allowing the definition of logical deployment packages groups to define Solutions and Services.

THE ASSURE PRINCIPLES

FULLY AUTOMATED



The whole Automation Management System

Release Service

Create Release / Version for
- Deployment Package
- Services
- Solutions

UPDATE Pipeline
- Artifact Promotion
- Deployment across Dev environments

PUSH Pipeline
- Build process
- QA
- Artifact Generation

PULL Pipeline
- Artifact Promotion
- Deployment across SDLC environments

Environment Service
- Manage Deploy Unit Definition
- Manage Environments
- Manage Deployments

Solution Composer
- Define Service
- Define Solution

Terraform API
- Execute Terraform

Create Deploy Package Version

Get Solution / Service Data

Trigger [2]

Request Deployment

Get Versions Data

Trigger [1]

Request Deployment

Get DU Data for Solution / Service Composition

Get Solution / Service Data

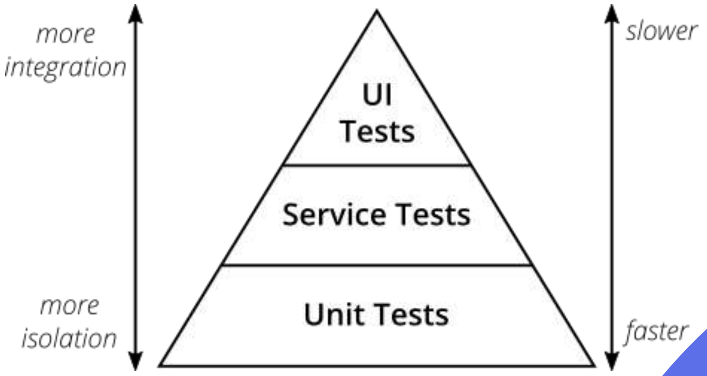
Request Deployment

1. *Through Artifactory*
2. *New DP version*

Tooling: DevCloud Jenkins, GitHub and Artifactory instances

> “IF YOU WANT TO GET SERIOUS ABOUT AUTOMATED TESTS FOR YOUR SOFTWARE, THERE IS ONE KEY CONCEPT YOU SHOULD KNOW ABOUT: THE TEST PYRAMID”

Mike Cohn /



THE ASSURE PRINCIPLES

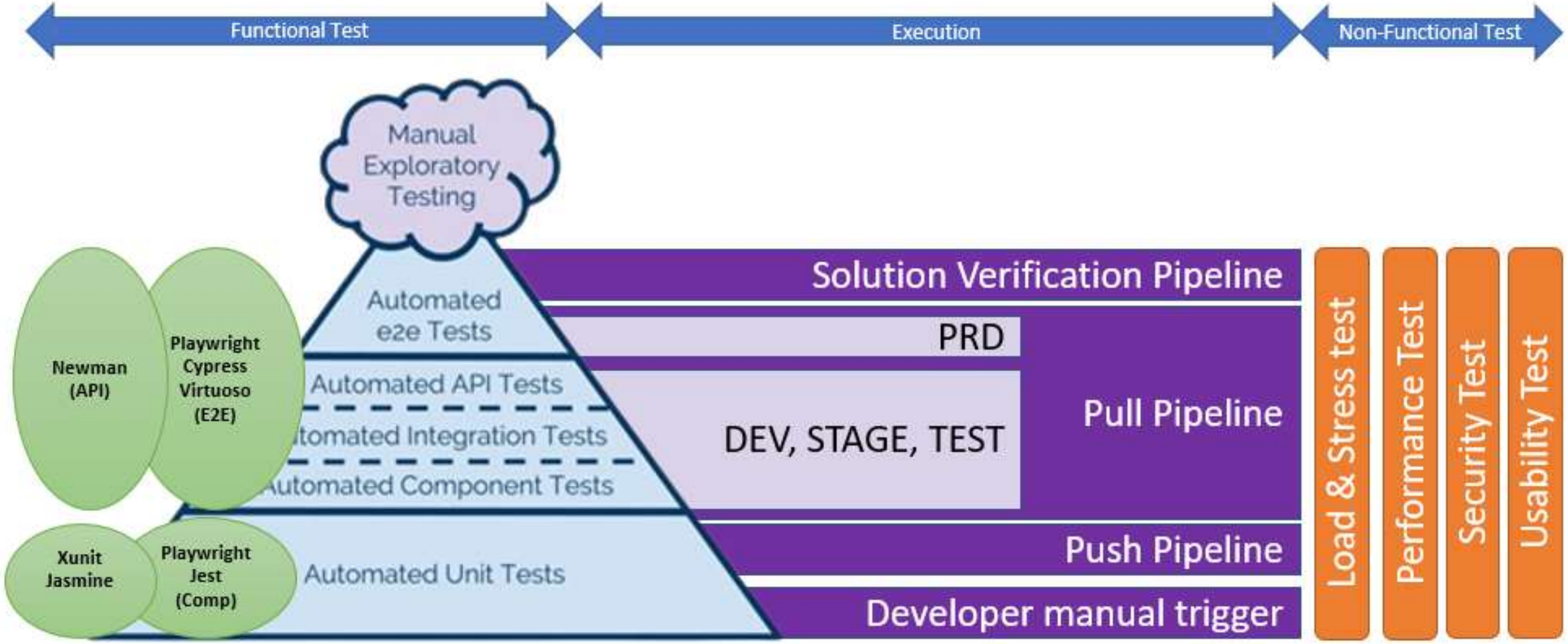
FULLY AUTOMATED

Assure testing Approach

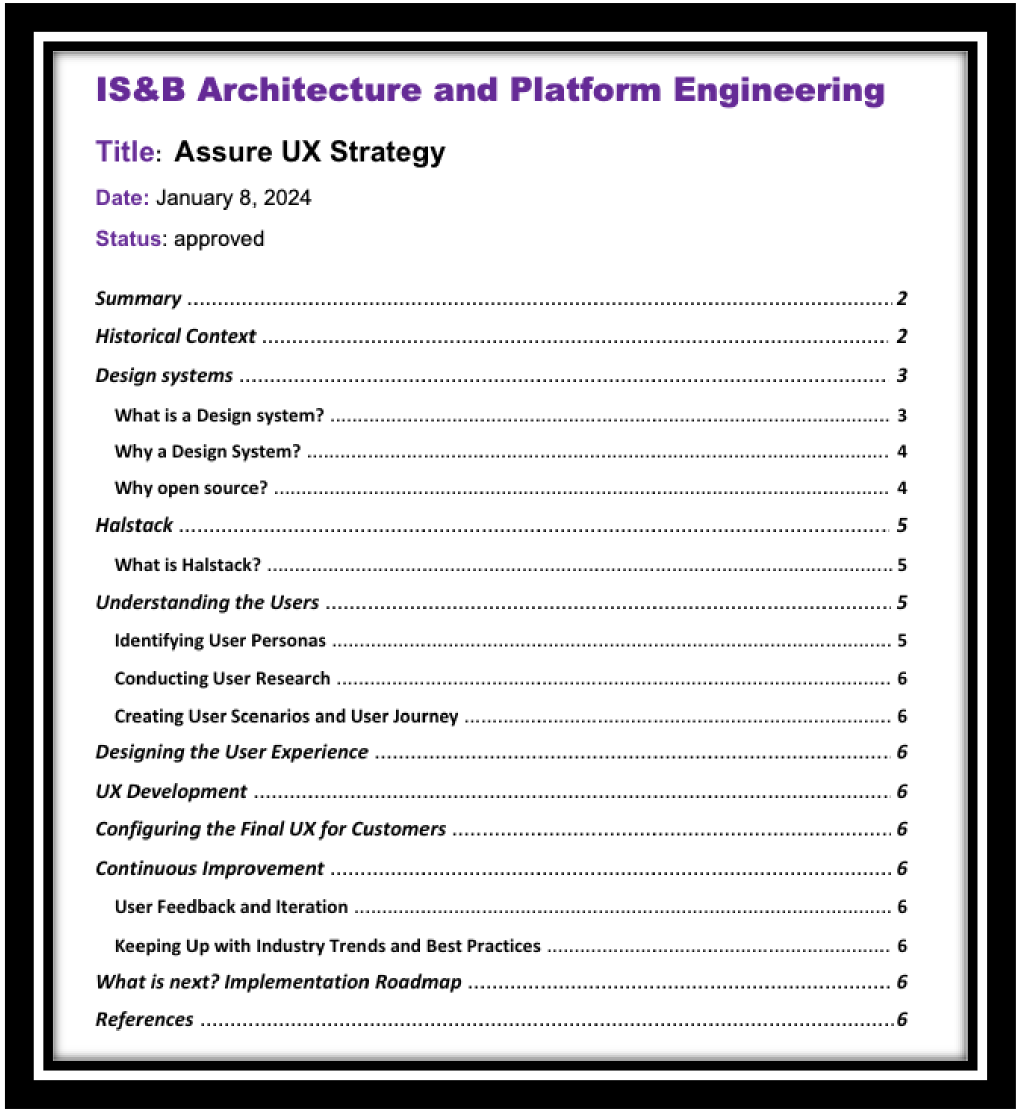
✓ Covering the entire SDLC Process

✓ Mapping different tools to the different test paradigms

✓ Shared responsibility: Developer/app Team writes the scripts; the Pipelines execute them for you



THE ASSURE PRINCIPLES UX DESIGN FIRST



IS&B Architecture and Platform Engineering

Title: Assure UX Strategy

Date: January 8, 2024

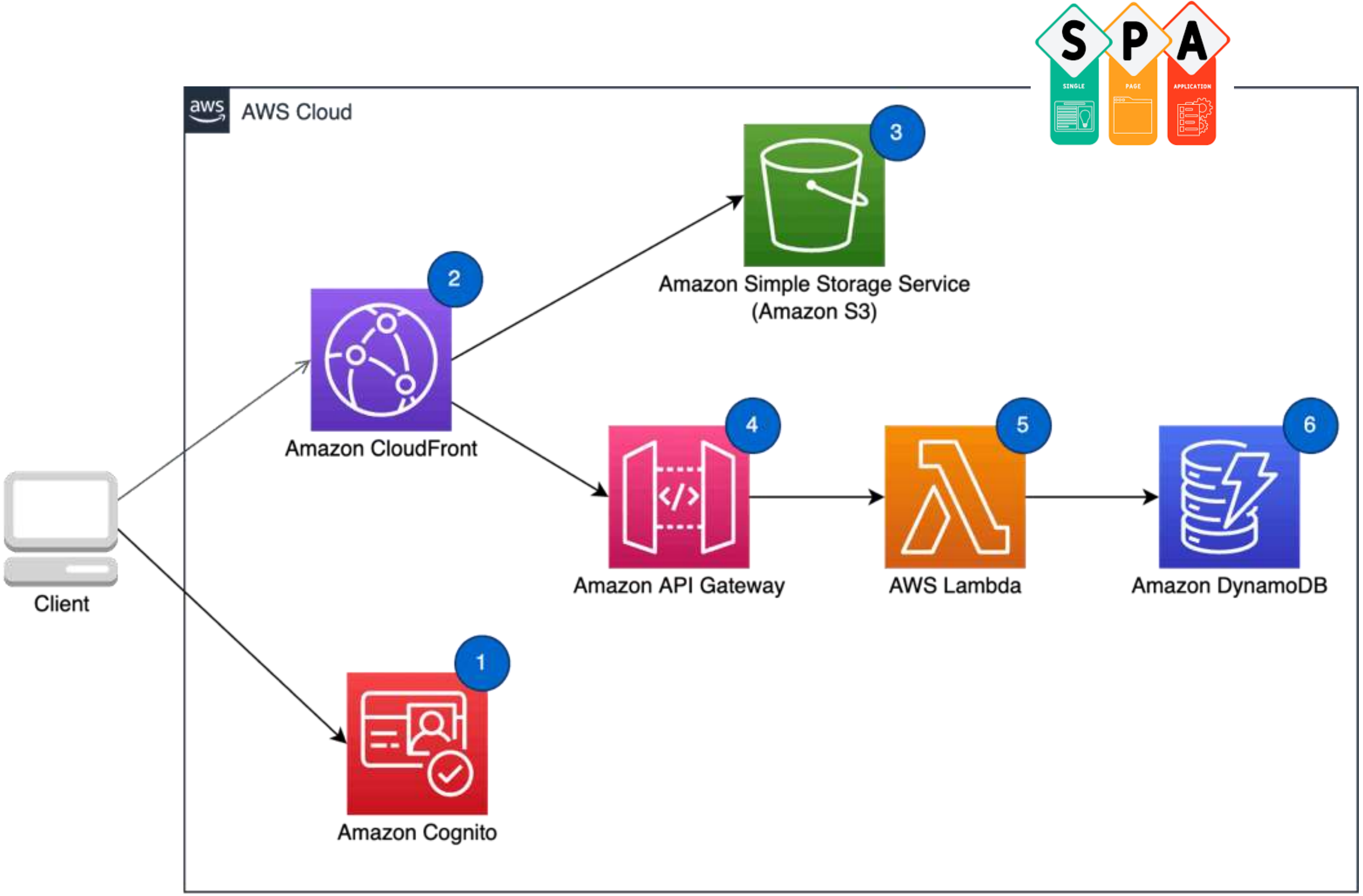
Status: approved

THE ASSURE PRINCIPLES

UX DESIGN FIRST

✓ Scalable and cost-effective

✓ Integrated with our IAM strategy

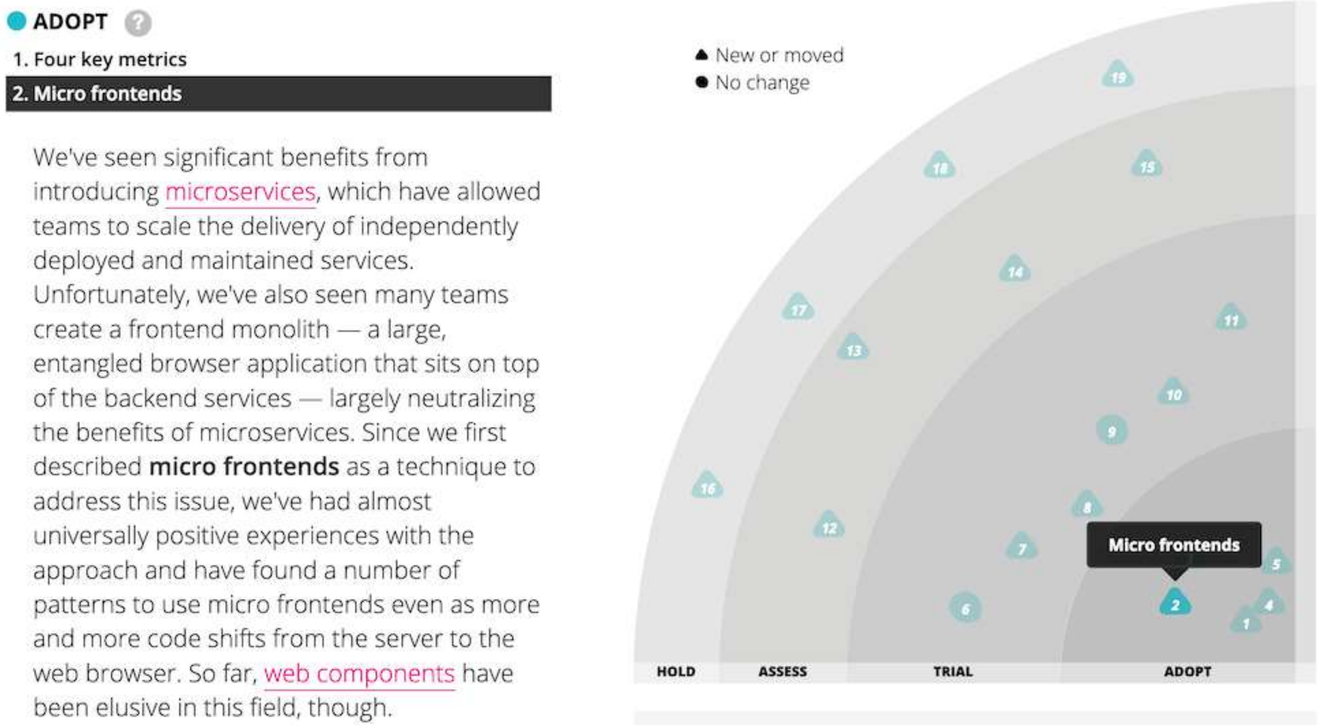


THE ASSURE PRINCIPLES

UX DESIGN FIRST

✓ Good developer experience: Modern frameworks + microfrontend architecture

✓ Composable



● ADOPT ?

1. Four key metrics

2. Micro frontends

We've seen significant benefits from introducing microservices, which have allowed teams to scale the delivery of independently deployed and maintained services. Unfortunately, we've also seen many teams create a frontend monolith — a large, entangled browser application that sits on top of the backend services — largely neutralizing the benefits of microservices. Since we first described **micro frontends** as a technique to address this issue, we've had almost universally positive experiences with the approach and have found a number of patterns to use micro frontends even as more and more code shifts from the server to the web browser. So far, web components have been elusive in this field, though.

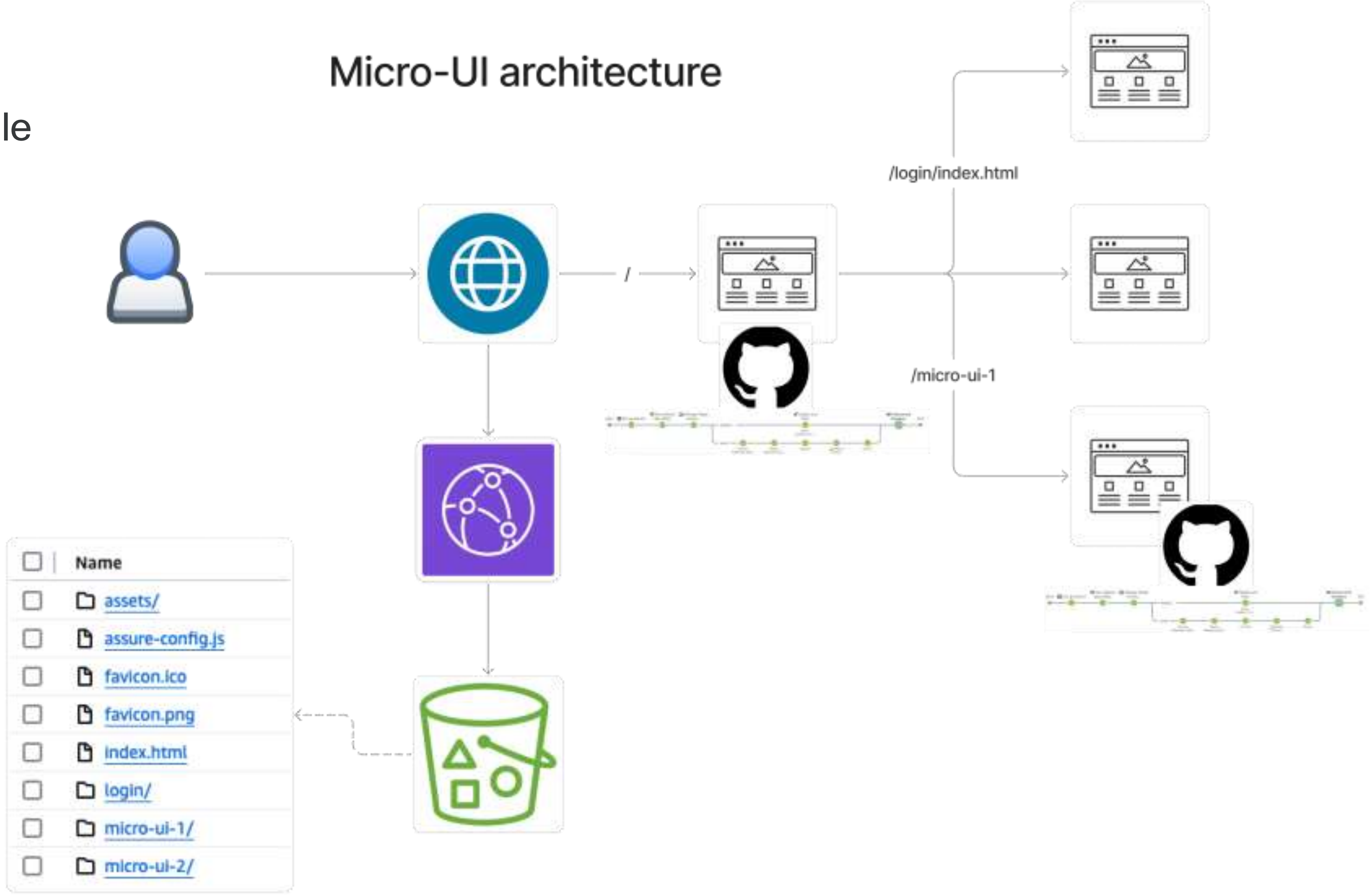
> *“An architectural style where independently deliverable frontend applications are composed into a greater whole”*

THE ASSURE PRINCIPLES

UX DESIGN FIRST

✓ Composable

Micro-UI architecture

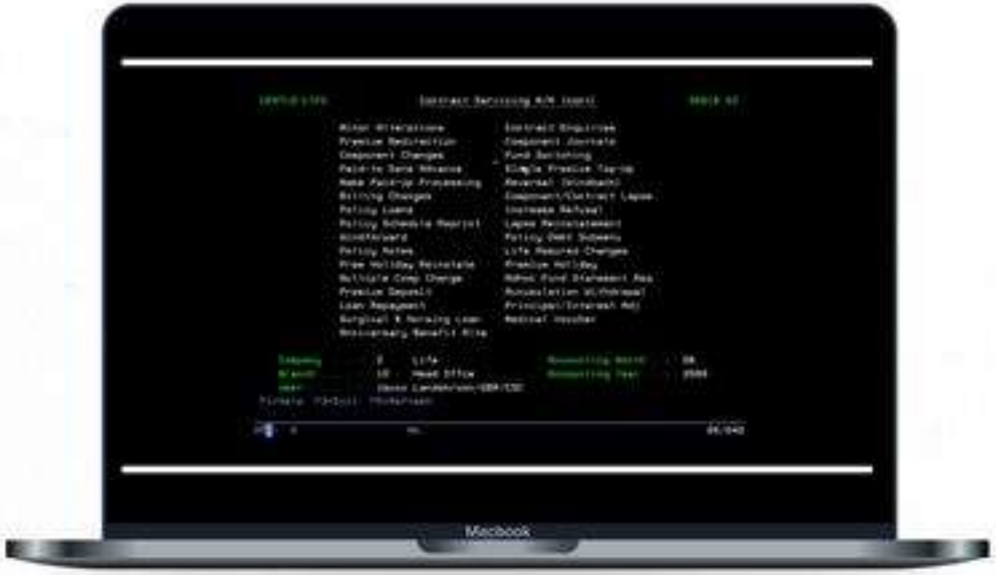


□	Name
□	assets/
□	assure-config.js
□	favicon.ico
□	favicon.png
□	index.html
□	login/
□	micro-ui-1/
□	micro-ui-2/

THE ASSURE PRINCIPLES

UX DESIGN FIRST

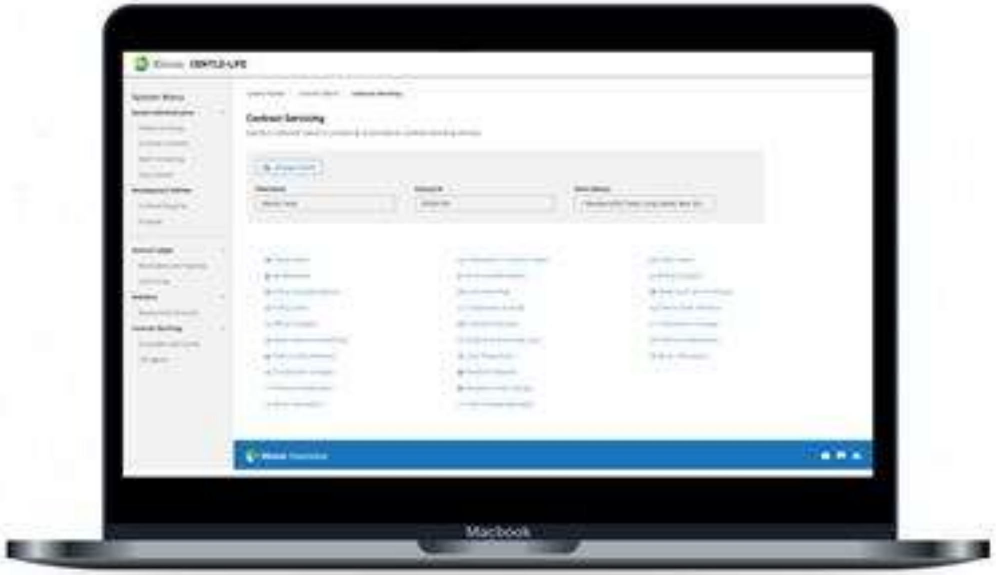
✓ Modern



```
[illegible]          Contract Servicing [illegible]          [illegible]

          Minor Alterations           Contract Enquiries
          Premium Redirection         Component Journals
          Component Changes           Fund Switching
          Paid-to Date Advance        Single Premium Top-Up
          Make Paid-Up Processing     Reversal (Windback)
          Billing Changes             Component/Contract Lapse
          Policy Loans                Increase Refusal
          Policy Schedule Reprint     Lapse Reinstatement
          Windforward                 Policy Debt Subsequ
          Policy Notes                Life Assured Changes
          Prem Holiday Reinstate      Premium Holiday
          Multiple Comp Change        AdHoc Fund Statement Req
          Premium Deposit             Accumulation Withdrawal
          Loan Repayment              Principal/Interest Adj
          Surgical & Nursing Loan     Medical Voucher
          Anniversary Benefit Rise

   Company        2    Life                 Accounting Month   :  08
   Branch         10   Head Office          Accounting Year    :  2004
   User                [illegible]
```

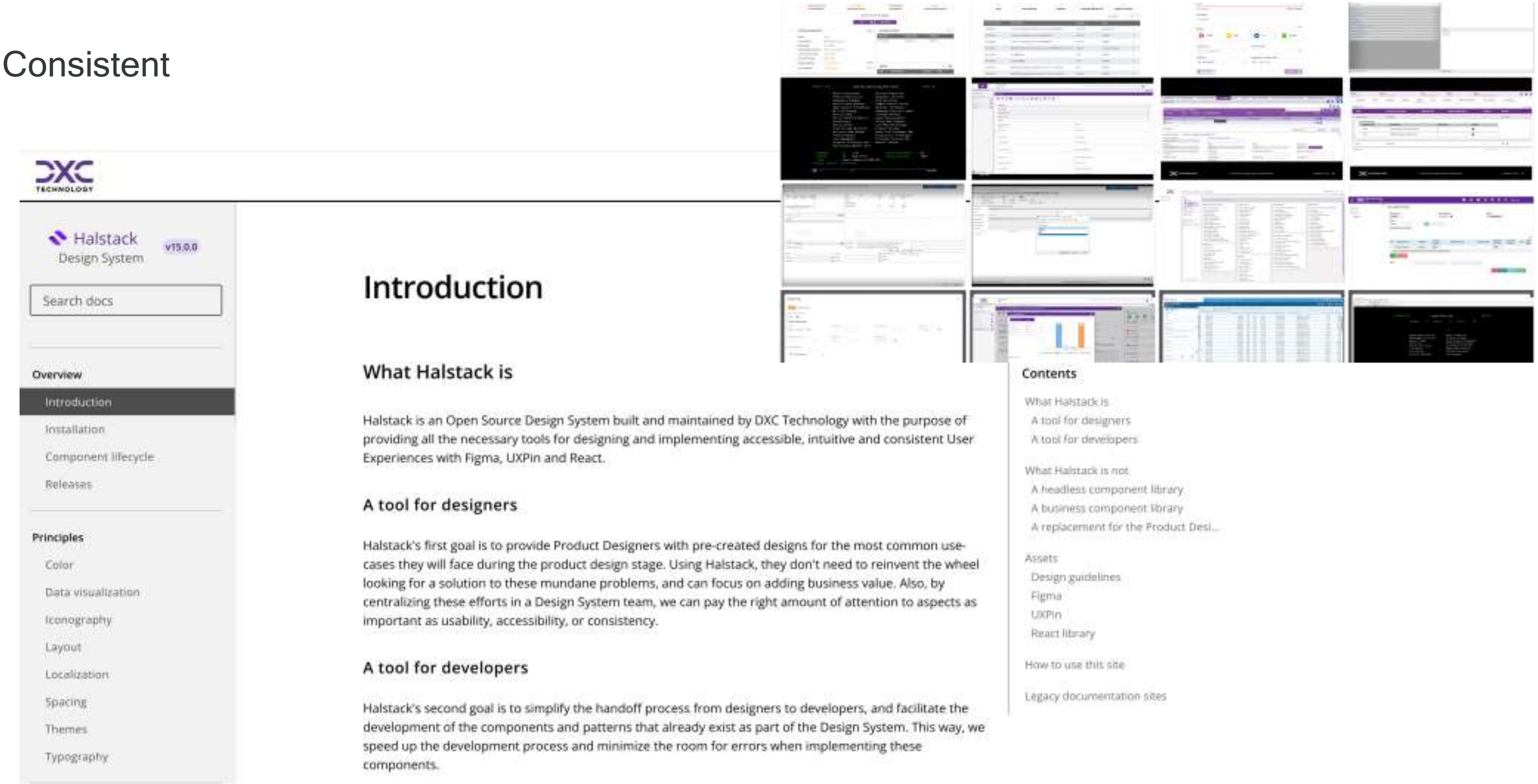


Contract Servicing

THE ASSURE PRINCIPLES

UX DESIGN FIRST

✓ Consistent



DXC TECHNOLOGY

Halstack Design System v15.0.0

Search docs

Overview

- Introduction
- Installation
- Component lifecycle
- Releases

Principles

- Color
- Data visualization
- Iconography
- Layout
- Localization
- Spacing
- Themes
- Typography

Introduction

What Halstack is

Halstack is an Open Source Design System built and maintained by DXC Technology with the purpose of providing all the necessary tools for designing and implementing accessible, intuitive and consistent User Experiences with Figma, UXPin and React.

A tool for designers

Halstack's first goal is to provide Product Designers with pre-created designs for the most common use-cases they will face during the product design stage. Using Halstack, they don't need to reinvent the wheel looking for a solution to these mundane problems, and can focus on adding business value. Also, by centralizing these efforts in a Design System team, we can pay the right amount of attention to aspects as important as usability, accessibility, or consistency.

A tool for developers

Halstack's second goal is to simplify the handoff process from designers to developers, and facilitate the development of the components and patterns that already exist as part of the Design System. This way, we speed up the development process and minimize the room for errors when implementing these components.

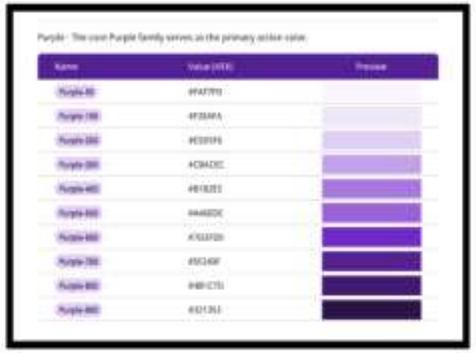
THE ASSURE PRINCIPLES

UX DESIGN FIRST

DRAFT

Style Guidelines

Updated: June 05th - 2024


Purple - The core Purple family serves as the primary action color.

Name	Value (HEX)	Preview
Purple-50	#FAF7FD	■
Purple-100	#F2EAFA	■
Purple-200	#E5D5F6	■
Purple-300	#C9ACEC	■
Purple-400	#B182E3	■
Purple-500	#A46EDE	■
Purple-600	[illegible]	■
Purple-700	#5F249F	■
Purple-800	#4B1C7D	■
Purple-900	#321353	■

Our selected typography helps in structuring our user's experience based on the visual impact that it has on the user interface content.

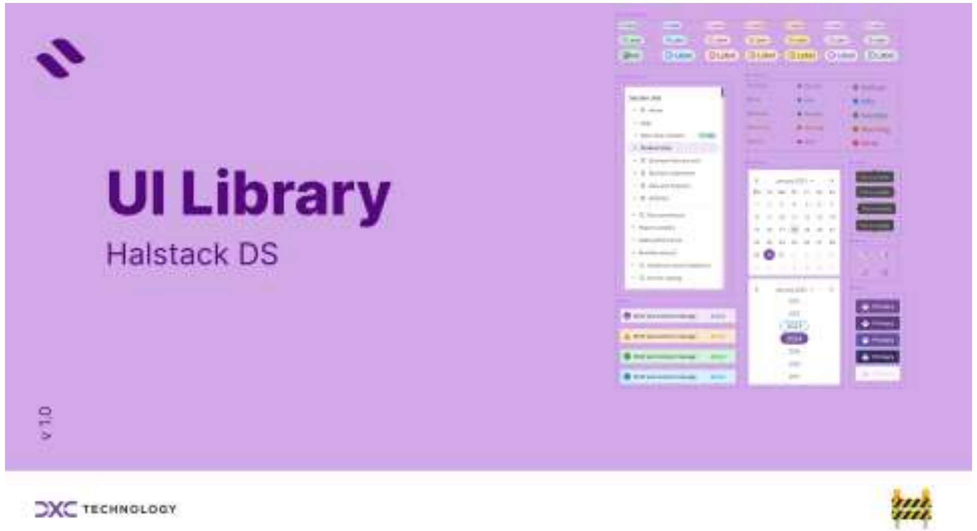
It defines what is the first noticeable piece of information or data based on the font shape, size, color, or type and it highlights some pieces of text over the rest. Some typographic elements used in Halstack Design System include headers, body, taglines, captions, and labels.

Typography Styles

Halstack Style Guidelines

CDK

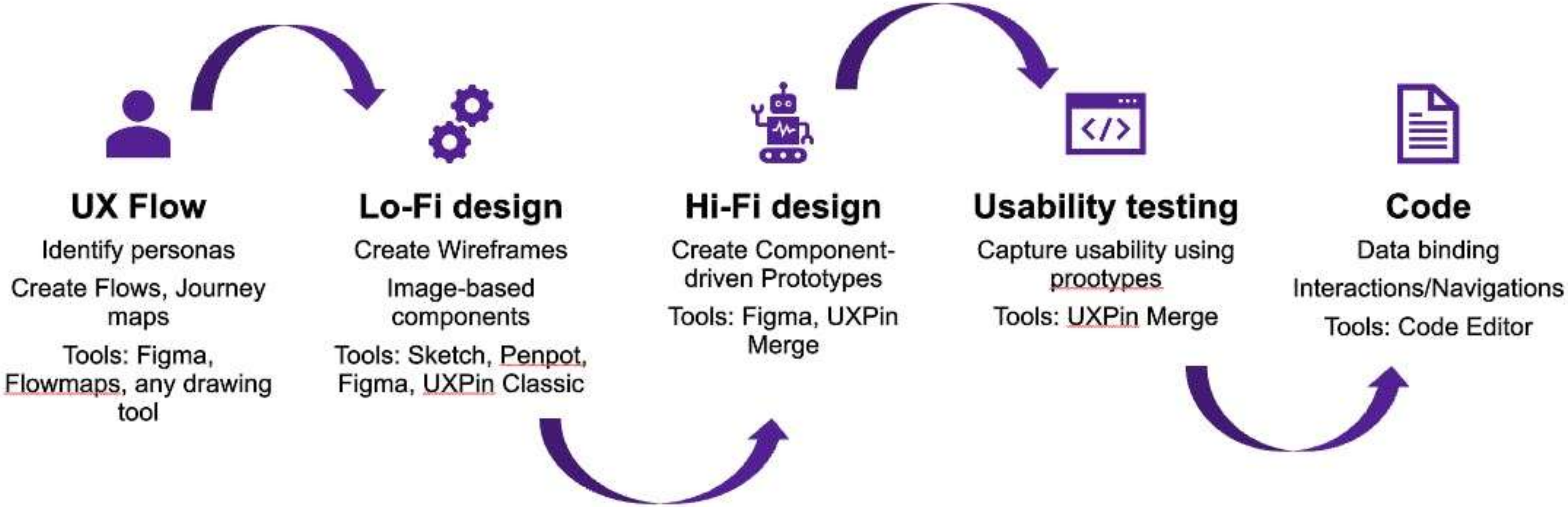
DXC TECHNOLOGY halstack-react


UI Library

Halstack DS

THE ASSURE PRINCIPLES
UX DESIGN FIRST

✓ Design First



UX Flow

Identify personas

Create Flows, Journey maps

Tools: Figma, Flowmaps, any drawing tool

Lo-Fi design

Create Wireframes

Image-based components

Tools: Sketch, Penpot, Figma, UXPin Classic

Hi-Fi design

Create Component-driven Prototypes

Tools: Figma, UXPin Merge

Usability testing

Capture usability using prootypes

Tools: UXPin Merge

Code

Data binding

Interactions/Navigations

Tools: Code Editor

ARCHITECTURE GOVERNANCE

Goal: Foster the adoption of the Internal Development Platform by all the product Teams to ensure business products can be delivered as SaaS and operated at scale by DXC for our Customers.

SVF

- The **S**oftware **V**alue **F**ramework represents the Assure Principles
- Founded by the Key Enterprise and system architecture decisions

1. API and Resource Model Architecture and Management
2. Cloud Deployment Architecture
3. Infrastructure as Code (IaC)
4. CI-CD Pipelines
5. CDP (Client Deployment Packaging)
6. AMS (Automation Management Services)
7. CDAM (Cyber Defence and Access Management)
8. Data Services and Data Pipelines
9. Event Service
10. UI/UX
11. SaaSOps
12. Standard Tech Stack

NUOUS ATION

self-evaluation process

npleted = Compliance

rogress = Planned for ure Release

ce Criteria: Assure 3oard (Mandates owners) review and approve

ARCH CE

Goal: Foster the ms to ensure business produc ustomers.

1 Implementation of compliant APIs and Resource Models
2 Solution/product deployable in containers (and/or serverless) on AWS
3 Infrastructure defined as Deploy Units (DUs) utilizing Platform Terraform modules
4 Implementation of Platform CI-CD Pipeline template (Push)
5 Environments, deployments and updates managed by AMS
6 CDAM Cyber Defense onboarding for production environments
7 CDAM IAM (Identity and Access Management) Integration\adoption
8 Following AI Guidelines
9 Defined 'Cost" model for infra and ops
10 Client configurations packaged using CDP and managed in AMS
11 OOTB data ingestion into DSDP
12 Implementation of ISB UI/UX Style Guidelines
13 Implementation of Platform Promotion CI-CD Pipeline template (Pull)
14 SaaSOps onboarding
15 Assure 360
16 Solution Properly Componentized
17 Event Driven
18 Supported Tech Stack

SVF

- The **S**oftware V represents the Assure Principles
- Founded by the Key Enterprise and system architecture decisions

MANDATES

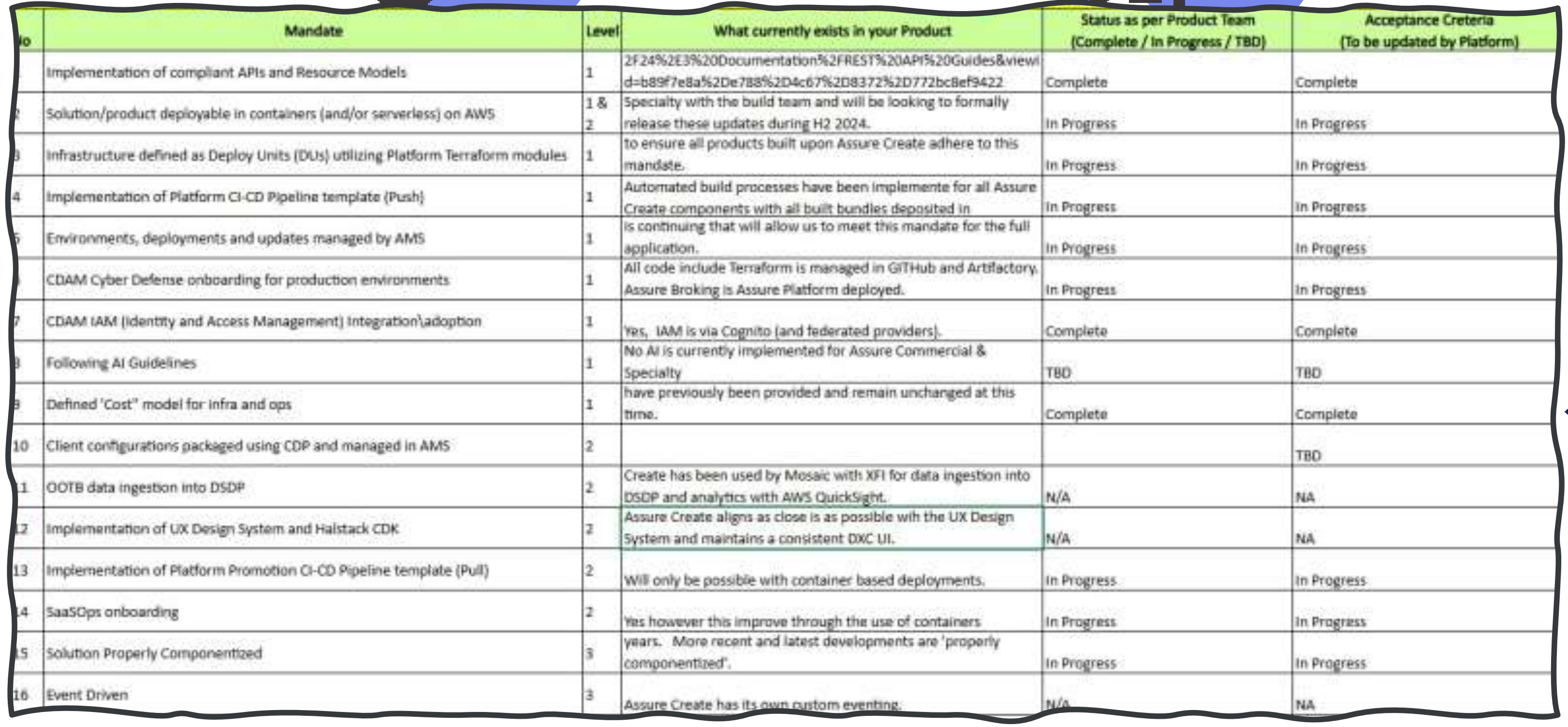
- List of requirements for the Assure Products
- Different levels of maturity

TINUOUS UATION

- Quarterly **self-evaluation process**
  - Completed = Compliance
  - In Progress = Planned for Future Release
- **Acceptance Criteria:** Assure Platform Board (Mandates owners) review and approve

ARCHITECTURE GOVERNANCE

Goal: Foster the adoption of the Internal Development Platform by all the product Teams to ensure business products can be delivered as SaaS and operated at scale by DXC for our Customers.



No	Mandate	Level	What currently exists in your Product	Status as per Product Team (Complete / In Progress / TBD)	Acceptance Creteria (To be updated by Platform)
1	Implementation of compliant APIs and Resource Models	1	2F24%2E3%20Documentation%2FREST%20API%20Guides&viewid=b89f7e8a%2De788%2D4c67%2D8372%2D772bc8ef9422	Complete	Complete
2	Solution/product deployable in containers (and/or serverless) on AWS	1 & 2	Specialty with the build team and will be looking to formally release these updates during H2 2024.	In Progress	In Progress
3	Infrastructure defined as Deploy Units (DUs) utilizing Platform Terraform modules	1	to ensure all products built upon Assure Create adhere to this mandate.	In Progress	In Progress
4	Implementation of Platform CI-CD Pipeline template (Push)	1	Automated build processes have been implemente for all Assure Create components with all built bundles deposited in	In Progress	In Progress
5	Environments, deployments and updates managed by AMS	1	is continuing that will allow us to meet this mandate for the full application.	In Progress	In Progress
6	CDAM Cyber Defense onboarding for production environments	1	All code include Terraform is managed in GITHub and Artifactory. Assure Broking is Assure Platform deployed.	In Progress	In Progress
7	CDAM IAM (Identity and Access Management) integration\adoption	1	Yes, IAM is via Cognito (and federated providers).	Complete	Complete
8	Following AI Guidelines	1	No AI is currently implemented for Assure Commercial & Specialty	TBD	TBD
9	Defined 'Cost" model for infra and ops	1	have previously been provided and remain unchanged at this time.	Complete	Complete
10	Client configurations packaged using CDP and managed in AMS	2			TBD
11	OOTB data ingestion into DSDP	2	Create has been used by Mosaic with XFI for data ingestion into DSDP and analytics with AWS QuickSight.	N/A	NA
12	Implementation of UX Design System and Halstack CDK	2	Assure Create aligns as close is as possible wih the UX Design System and maintains a consistent DXC UI.	N/A	NA
13	Implementation of Platform Promotion CI-CD Pipeline template (Pull)	2	Will only be possible with container based deployments.	In Progress	In Progress
14	SaaSOps onboarding	2	Yes however this improve through the use of containers	In Progress	In Progress
15	Solution Properly Componentized	3	years. More recent and latest developments are 'properly componentized'.	In Progress	In Progress
16	Event Driven	3	Assure Create has its own custom eventing.	N/A	NA

CONTINUOUS EVALUATION

- Quarterly **self-evaluation process**
  - Completed = Compliance
  - In Progress = Planned for Future Release
- **Acceptance Criteria:** Assure Platform Board (Mandates owners) review and approve

LESSONS LEARNT

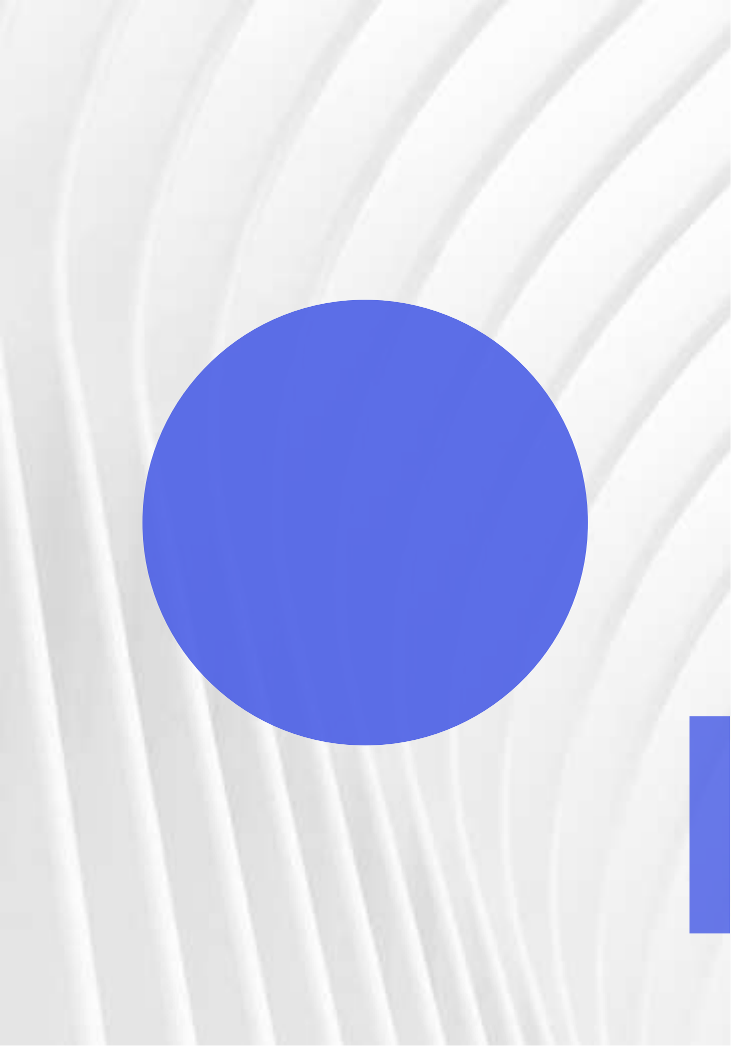
ADRs and IDPs can’t solve culture

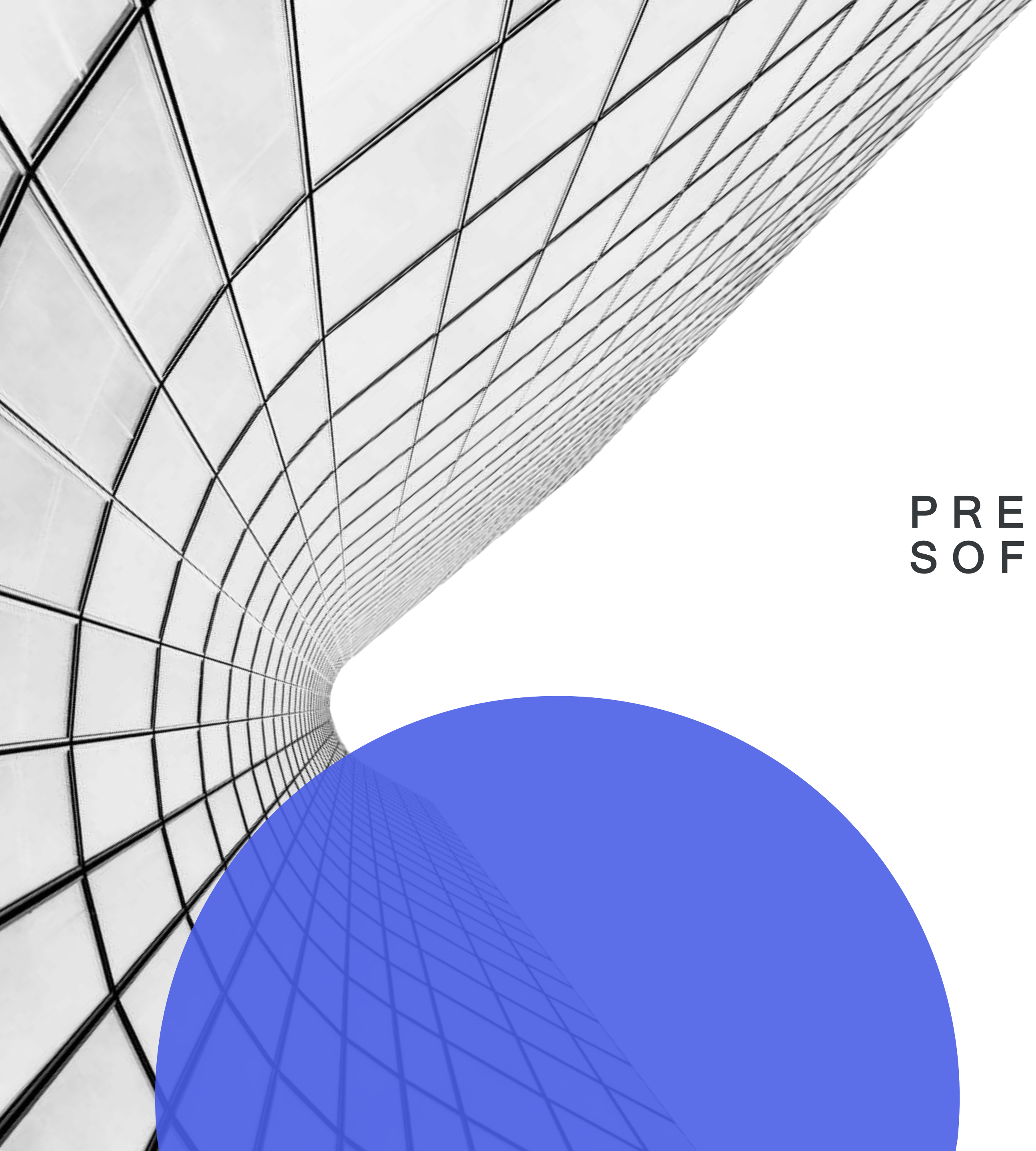
Guidance vs Governance

High quality software is cheaper to produce and keeps your team happy

Timing is important (careful with what you build)

Good architecture decisions don't always win





PRESENT AND FUTURE OF SOFTWARE ARCHITECTURE

CONTINUOUS LEARNING AND INNOVATION

It is on the DNA of a software architect to keep updated on the latest market trends about tools, techniques, platforms, languages and frameworks. The knowledge and experience about that will influence strongly on the elaboration of ADRs, a key element while designing, in general, any System.

Main sources (but not all)

General Technology Trends
(thoughtworks.com/radar)

- ✓ Quadrants Technologies
- ✓ Maturity Assessment
- ✓ Practical Industry Experience
- ✓ Regularly Updated Trends
- ✓ Vendor-Neutral Approach

AWS ecosystem
(multiple sources)

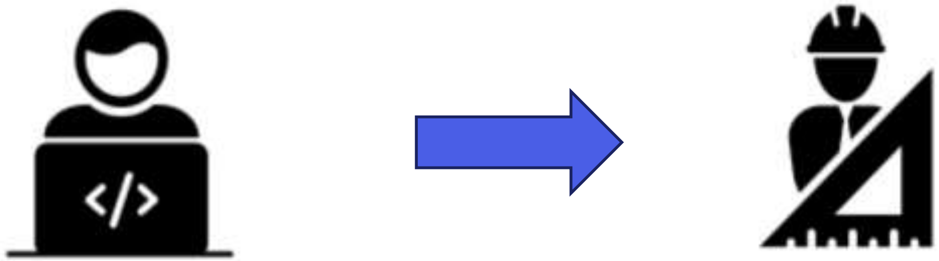
- ✓ Official Resources
- ✓ Community and Learning Platforms
- ✓ Technical Newsletters & Blogs
- ✓ Podcasts
- ✓ Hands-on Learning & Experimentation

Javascript ecosystem
(stateofjs.com/)

- ✓ Developer Preferences
- ✓ Trends & Adoption
- ✓ Frontend, Backend & Testing
- ✓ Data Visualization
- ✓ Community-Driven

Platform ecosystem (platformcon.com/)
✓ Focus on Platform Engineering - ✓ Talks by Experts - ✓ On-Demand & Free - ✓ Community-Driven

YOU BUILD IT, YOU SHIP IT



Traditional Model

- Developers focused only on application logic
- Architects and Ops controlled deployments
- Slow feedback loops, fragmented knowledge
- Learning opportunities were limited

Modern Software

- Developers Now Own Deployment & Operations
- Cloud Commodities make Architecture More Accessible
- Shift-Left Approach:
    - Developers Solve Architectural Problems Earlier

AI & SOFTWARE ARCHITECTURE

Traditional Model

- Manual analysis, best practices, and experience-based intuition to design systems
- Fixed patterns like microservices vs monolith, event-driven vs api

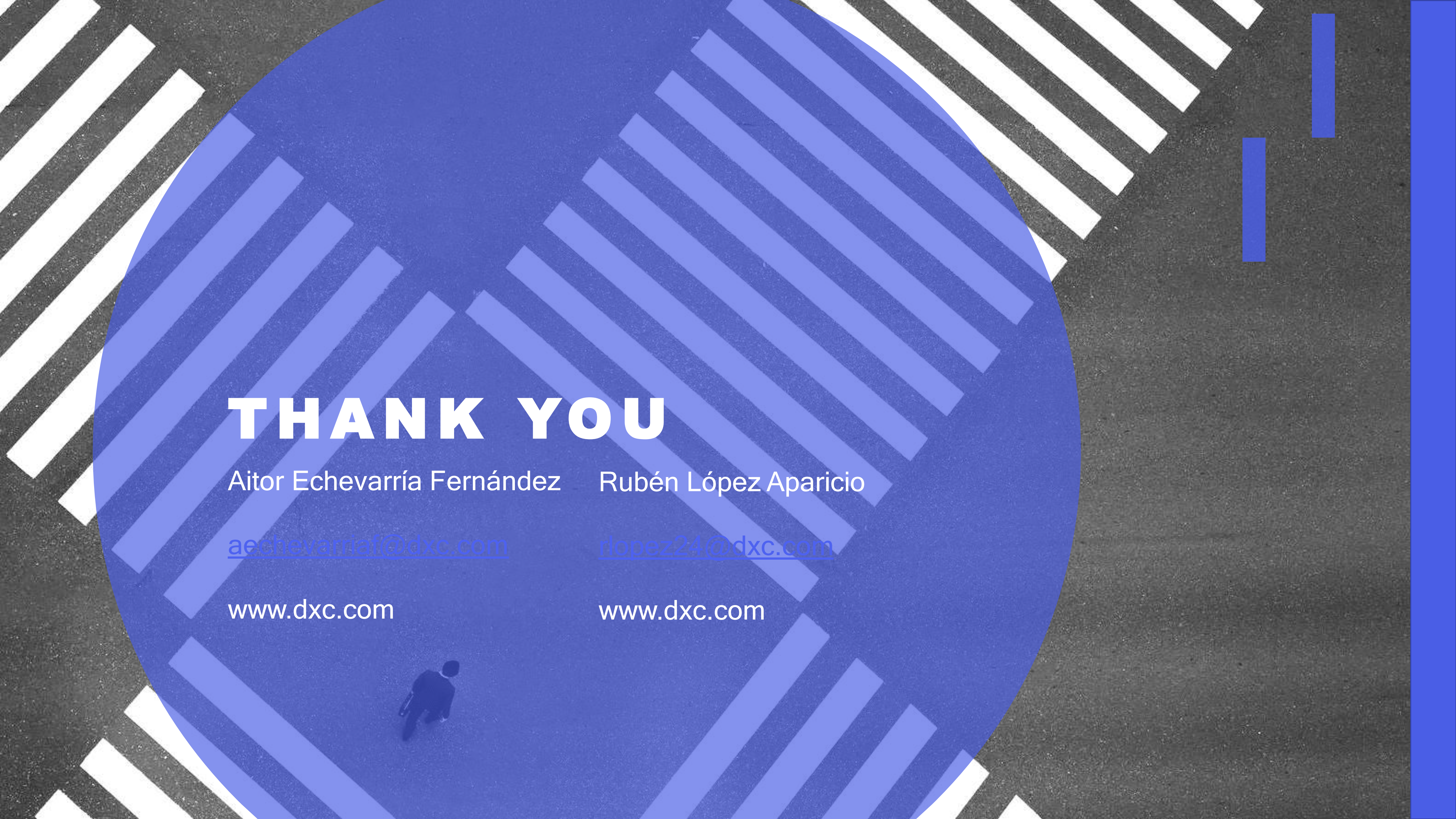
With AI

- **AI-powered architecture assistants:**
  - AWS Well-Architected AI
  - DataDog AI, Dynatrace AI
- **AI is Accelerating Exploration**
- **AI monitoring systems behaviour (AIOPs)**

FINAL TIPS

1. Take a second look at the continuous learning slide materials
2. Follow architecture Advocates in Social media, also local evangelist like Pablo Bermejo and Enrique Riesgo
3. Join local groups – AWS Asturias User Group

aws
User Group
Asturias, Spain



THANK YOU

Aitor Echevarría Fernández

aechevarriaf@dxc.com

www.dxc.com

Rubén López Aparicio

rlopez24@dxc.com

www.dxc.com