THE BEST RUN SAP

Lean Enterprise Architecture Toolkit Templates

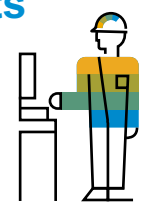
version 1.0 | February 2021 | PUBLIC

Table of Contents

Introduction

Combining Design Thinking and Architectural Thinking

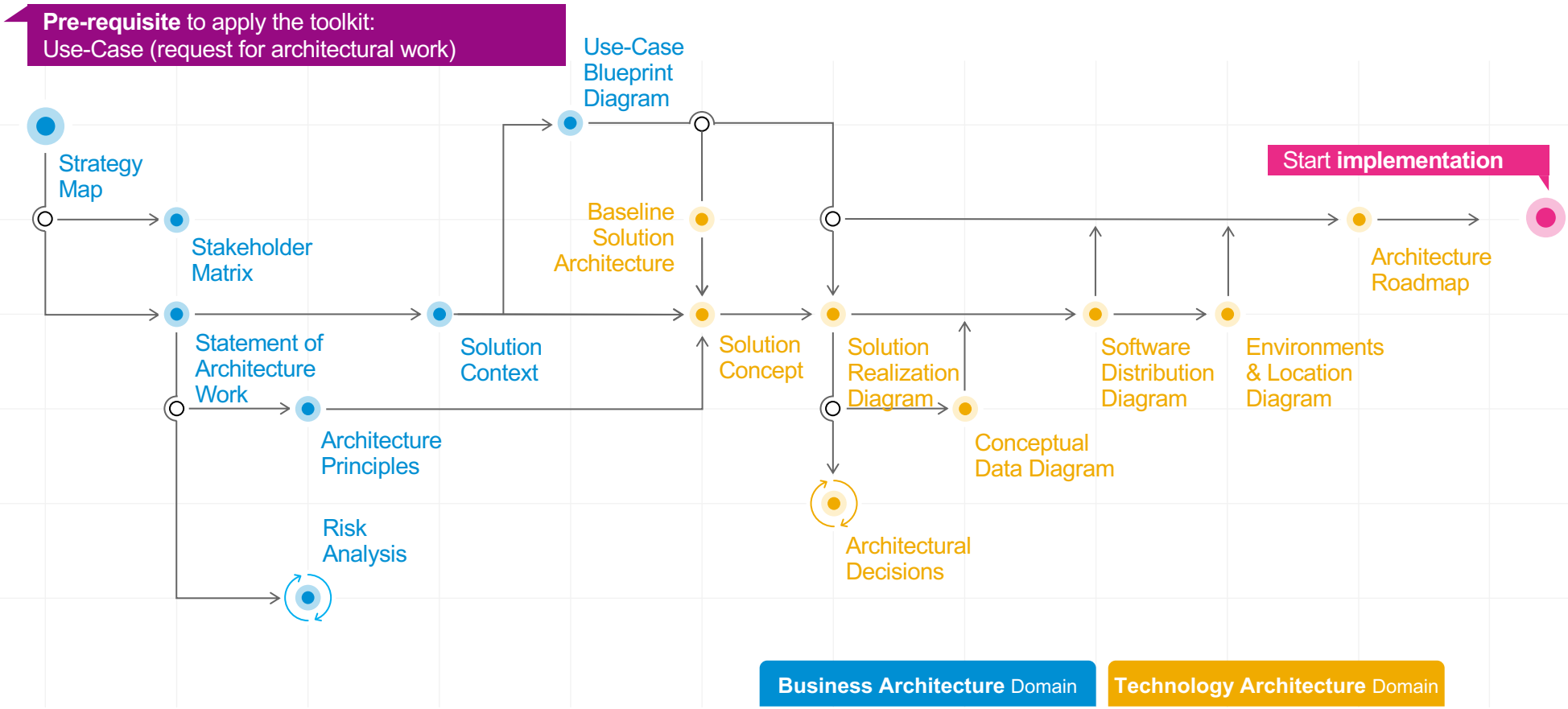
The Lean Enterprise Architecture Toolkit can be used in combination with the Human Centered Innovation Approach as defined by the SAP AppHaus. While the Lean Enterprise Architecture Toolkit applies an engineering mindset to understand, describe, and solve a business problem, the Human Centered Innovation Approach applies a user-centric view to the business problem with the help of Design Thinking. While Design Thinking focuses on the alignment of users and business, the Architectural Thinking with the toolkit focuses on the alignment of business and IT.

Phases	Explore	Discover	Design	Deliver
Design Thinking 	**Explore innovation opportunities:** ▪ Cluster of innovation opportunities ▪ Prioritize and select innovation opportunity ▪ Use-case definition ▪ Stakeholder map	**Gain common understanding of current environment and needs via consumer, employee, and market research:** ▪ First insights to overcome the challenges ▪ Research summary ▪ As-is user journey	**Design and create a prototype of the solution:** ▪ Storyboard for selected solution idea ▪ Low-Fidelity sketches ▪ Clickable prototype ▪ Visual design ▪ User stories	**Develop and deliver the tailored business and technical solution for productive use:** ▪ Deliver productive solution ▪ Customer acceptance report
	Use-Case Blueprint Diagram			
Architectural Thinking Work Products 	**Understand your company's business goals and strategic drivers**: ▪ Strategy Map ▪ Stakeholder Map ▪ Statement of Architecture Work	**Develop a target business architecture that describes how your company needs to operate to achieve business goals**: ▪ Solution Context ▪ Architecture Principles ▪ Risk Analysis	**Develop a target technical architecture that addresses the statement of architecture work and stakeholder concerns:** ▪ Baseline Application Architecture ▪ Solution Concept ▪ Solution Realization ▪ Conceptual Data Diagram ▪ Software Distribution Diagram ▪ Architectural Decisions	**Define an architecture roadmap based upon gaps between baseline and target architecture:** ▪ Architecture Roadmap ▪ Environments & Location Diagram
Domains	**Business Architecture**		**Technical Architecture**	

Overview of the content of the Lean EA Toolkit

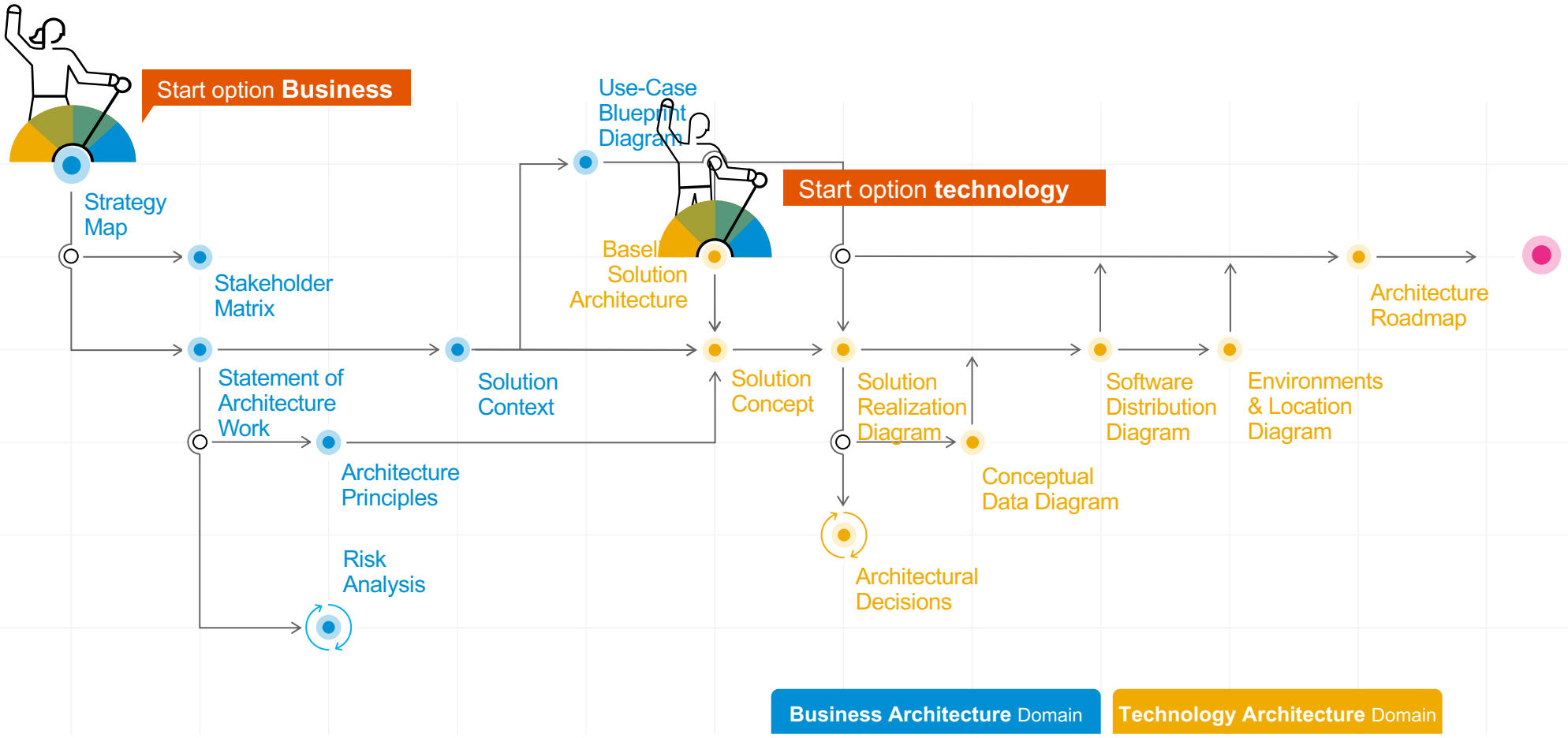
The Lean Enterprise Architecture Toolkit is comprised of 15 work products: 7 work products describe the business domain, and 8 work products describe the technical domain. Work products from the business domain are used to communicate with stakeholders from the business departments as well as business executives and business management. Work products from the technical domain are predominantly used to communicate with stakeholders from IT. Each work product of the toolkit is described by a template, an example showing the adoption of the template, and a brief guide for creating the work product based on its template.

Generally, the work products of the toolkit are applied in a defined sequence. The reason for this is, that results from one work product serve as input for the creation of other work products.



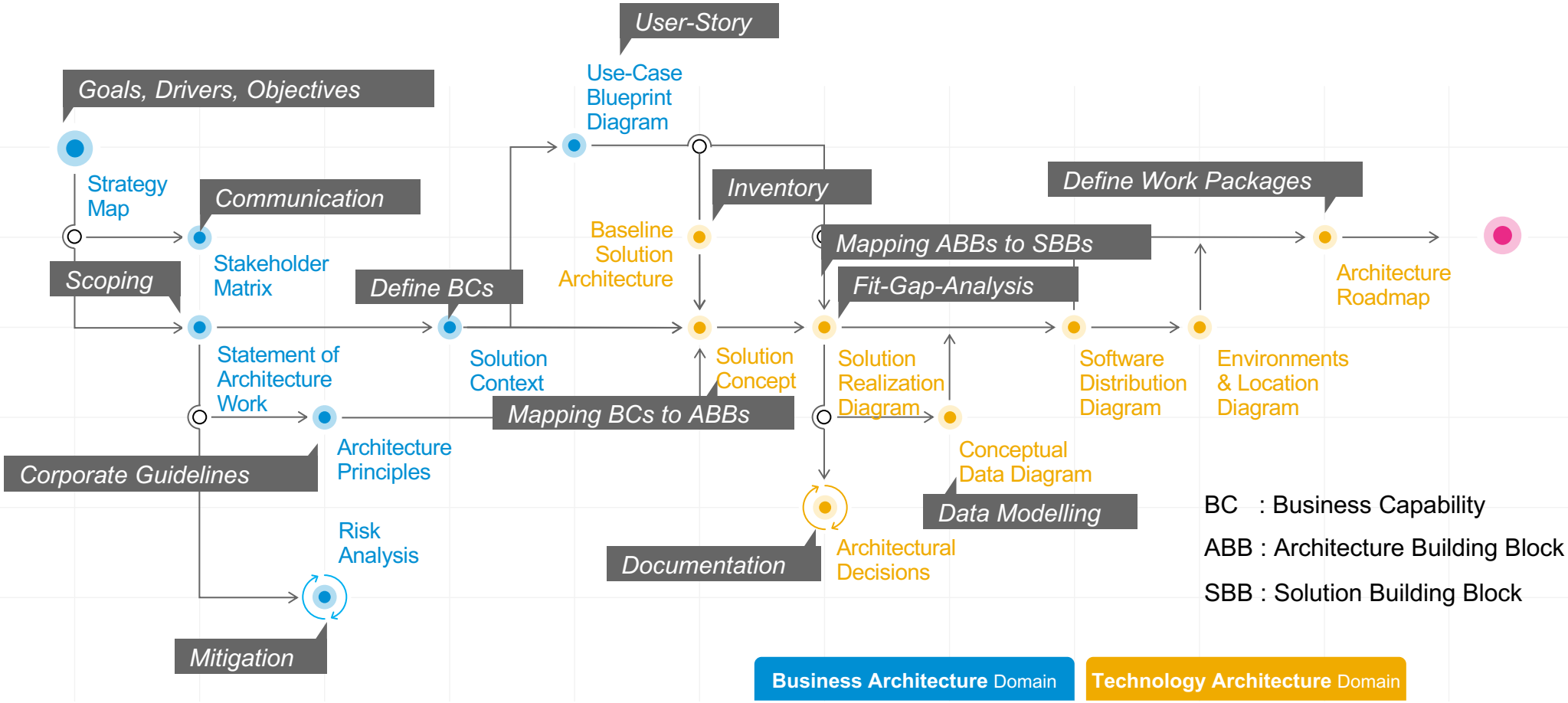
Start options for using the Lean EA Toolkit

Depending on the driver for architectural work, the sequence of using the templates of the toolkit can be adapted, respectively. Considering a business department that issues a request of architectural work, the Strategy Map or Statement of Architecture Work might be one of the first work products being created. In case the IT department would like to discover the business value of using a specific technology, the Solution Concept or Baseline Solution Architecture might be the work products to start with the architecture development.



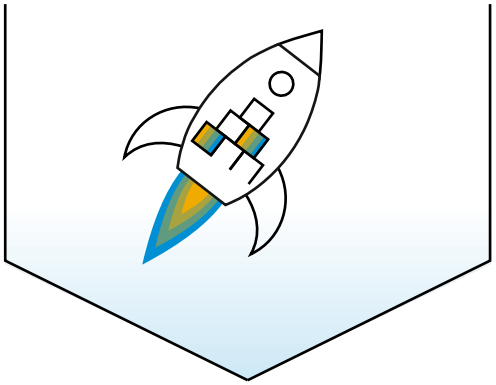
Activities when using the Lean EA Toolkit

Using the Lean EA Toolkit involves different activities for the creation of the work products. Each work product offers a different view on the architecture. It is the sum of all work products describing the architecture of the aspired solution.



Scenario used for sample work products

The samples provided for the Lean EA toolkit work products are based on a fictitious company named Rocket Chips Inc. The sample work products introduced in the following chapters assume a request of architecture work to improve the budget review and approval process for new R&D initiatives and business development projects at Rocket Chips.



Company name:
Rocket Chips Inc., a global semiconductor company founded in 2013, engaged in the design and fabrication of semiconductors that are specialized to run machine learning algorithms.

Tagline: *We bring you beyond!*

Drivers:

- Become global market leader in semiconductors optimized for machine learning algorithms
- Grow into new market segments by offering additional services

Goals:

- Grow fast, grow globally
- Fast and effective deployment of capital for business initiatives and research & development projects to support massive growth as well as innovation and adaptability (sources of competitive advantage)

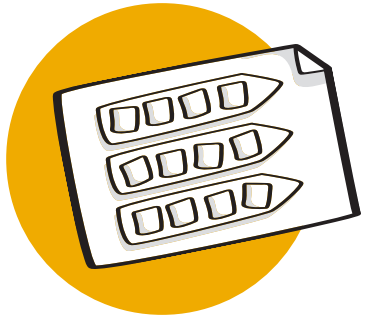
Explore Templates

Business Architecture Domain

Strategy Map

Instructions | Template | Example

Understand the business impact of your architectural work.



Strategy Map

Visualize the strategic intentions of your organization through goals, drivers, and objectives.

Understand Strategy

Gather the strategic intentions of your company and your organization.

Understand Goals

Gather the strategic goals of your organization and link related drivers, goals and objectives.

Associate your Use-Case

Identify the strategic goal(s), driver(s), and objectives that are supported by your architecture project.*

*This can be done at a later stage, when you have more details about your project.

Strategy Map
Instructions

Duration

approx. 30-90 minutes

Input

- **Corporate Strategy**
- **Annual Report**

Why & What

A Strategy Map is a graphical representation of the organization's goals and objectives. It visualizes the strategic intentions of the organization and de-composes it in related drivers, goals and objectives.

The Strategy Map provides context for your architecture work.

How to use it

1. Note the vision (purpose) of the company.

2. Document external or internal drivers that motivate the company to define its goals. Think about external factors such as market conditions or stakeholder demands. There might be more than one driver for the company or organization.

3. Document the corporate goals that influence the direction where the company is heading.

4. Add strategic goals which are specific goals the company or departments of the company want to achieve. Every strategic goal has associated objectives, that are specific, measurable and time bound. Also, add existing projects or activities that are supporting a strategic goal.

Tips & Tricks

Driver: An external or internal condition that motivates the organization to define its goals, such as customer and market behavior, competitive forces, legislation, etc.

Goal: A formulation of a (strategic) intention or (strategic) direction of the organization.

Objective: Specific, Measurable, Attainable, Realizable and Time bounded (S.M.A.R.T.) formulation of a (strategic) goal, motivates the requirements for a Capability

Depending on your initial understanding of the problem domain, and the details you got from the request of architectural work, you might already be in the position to clearly identify strategic goals and objectives that are specifically supported by your architectural work.

Strategy Map Template

Vision

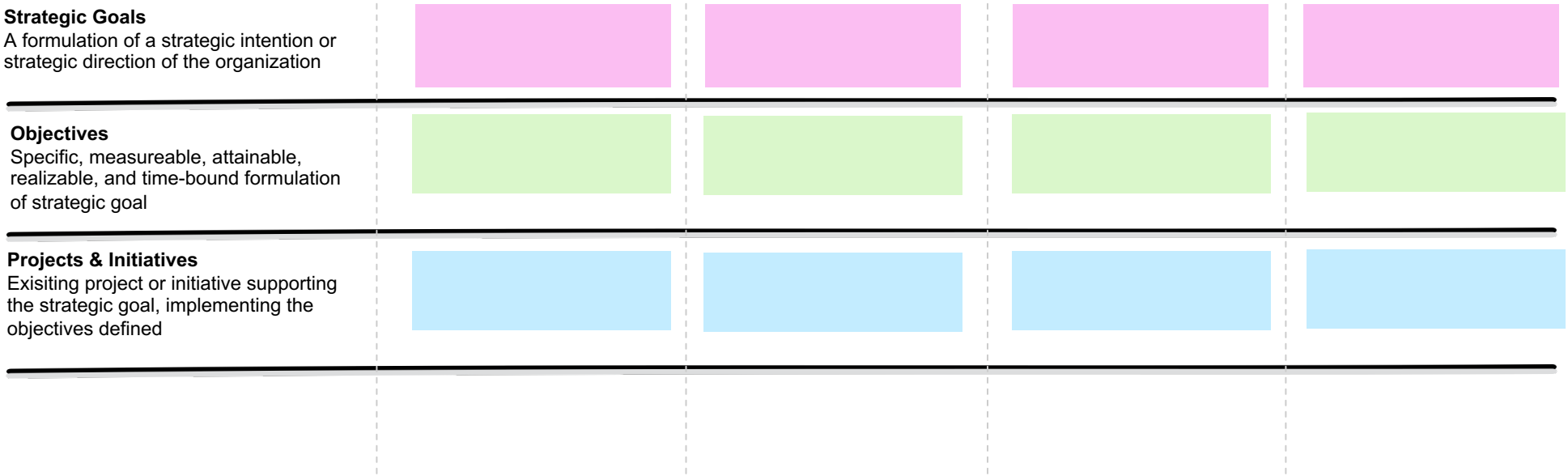
Add company's vision statement

Driver(s)

An external or internal condition(s) that motivate(s) the organization to define its goals, such as customer and market behavior, competitive forces, legislation, etc.

Corporate Goal(s)

Add a formulation of intention(s) or direction(s) of the company or organization.



Strategy Map Example

Rocket Chips Inc.

Vision

Create world-changing technology that enriches people's lives.

Driver

Unleash the potential of artifical intelligence by reliable, sclable,trusted data processing inspired by quantum computing.

Corporate Goal

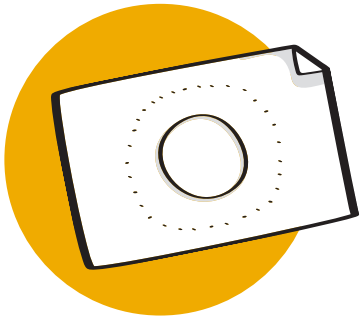
Global market leader in technology optimized for machine learning algorithms.

Strategic Goals	Fast and effective approval of business initiatives and research & development projects to support massive growth as well as innovation and adaptability	Engineer solutions for our customers' success with reliable, cloud-to-edge computing inspired by quantum technology	Provide new digital services supporting AI as a service	Sell digital services to customers via a marketplace. Offer API Hub to consume services
Objectives	▪ Reduce average time of approval from 25+ days to 2 days ▪ Automatic processing of at least 25% of requests with confidence	▪ Improve training speed of massive data sets by 75% until 2022 ▪ Reduce power consumption by 50% until 2022	Offer AI based services via APIs out of own data centers	A marketplace will be created to so sell services to stakeholders
Projects & Initiatives	▪ Global Growth Initiative 2025	▪ Quantum Flagship Initiative	▪ Digital Agenda 2023 ▪ Data Center Strategy	▪ Digital Agenda 2023 ▪ Project STELLAR

Stakeholder Matrix

Instructions | Template | Example

Understand key stakeholders for your architectural work and manage support for your architecture.

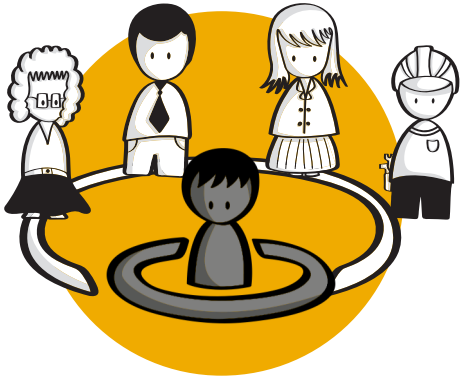


Stakeholder Matrix

What are the Stakeholders involved in the architecture? How do you engage with the stakeholders?

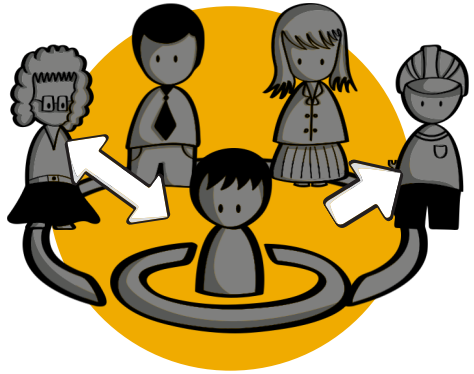
Identify stakeholders

Who are users affected by the aspired solution? Who has influence on your architecture? Who is interested in its success?



Understand stakeholders

Analyze and categorize your stakeholders. Who needs to support you? Who has potential to disrupt? Who needs to be informed?



Manage stakeholders

Based on your analysis, define your approach to stakeholder management. How do you regularly engage with your stakeholders?

Stakeholder Matrix

Instructions

Duration

approx. 30-60 minutes

Input

- Business Units
- Roles
- Users

Why & What

The purpose of stakeholder management is to ensure support for your architecture and improve its quality by addressing the concerns of your stakeholders.

You use stakeholder-specific architecture views being created with the toolkit to adequately communicate your architecture.

How to use it

1. Based on the use-case (request for architecture work), identify users, business units, parts of your organization, or a board area that are affected by the architecture or can influence your architectural work. List all stakeholders that are interested in the success of your architecture.

2. Understand each stakeholder's interest and concerns.

3. Derive the type of engagement you would like to have with the stakeholders. Do you need regular meetings every week to discuss the status of your architecture? Or is a monthly update via email enough?

Tips & Tricks

Think internal & external.

Use attributes like "*Key Player*", "*Keep satisfied*", "*Keep informed*", and "*Minimal effort*" to categorize stakeholder engagement. Your communication and interaction with a key player is pro-active and very regular. You want to make sure that this stakeholder is always informed, included in important decisions and regularly updated. Your interaction with a stakeholder with minimal effort is more of a reactive style, instead.

You can also decide which work products of the Lean EA toolkit are of interest for a specific stakeholder and you want to share, respectively. Associating work products with stakeholders can also be done at a later stage in your architecture development process.

Stakeholder Matrix Template

Stakeholder	Concern(s)	Engagement	Work Products *
Name and corporate function of the stakeholder	*Describe stakeholder's interests and concerns*	*Define engagement type: key player, keep satisfied, keep informed, minimal effort*	*Your work products that are of stakeholder's interest*

Source: TOGAF Standard, Version 9.2

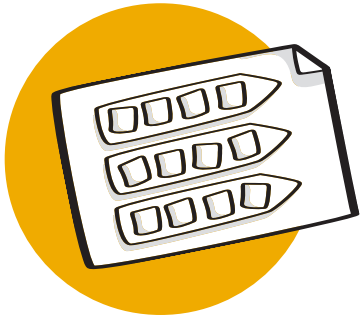
*Can be added at a later stage, when you have more details about your project.

Stakeholder Map Example

Stakeholder	Concern(s)	Engagement	Work Products
Paul Jung (CEO)	Understand how IT helps to advance business by supporting company's goals and objectives.	Keep satisfied	▪ Strategy map ▪ Statement of Architecture Work
Julie O'Brian (CFO)	Enterprise-level adoption of automation, leveraging analytics and connecting with other business units.	Key player	▪ Strategy map ▪ Statement of Architecture Work ▪ Solution context
An Liu (Director Business Development)	Identify and successfully deliver projects that implement growth opportunities.	Key player	▪ Statement of Architecture Work ▪ Solution concept

Statement of
Architecture Work

Instructions | Template | Example



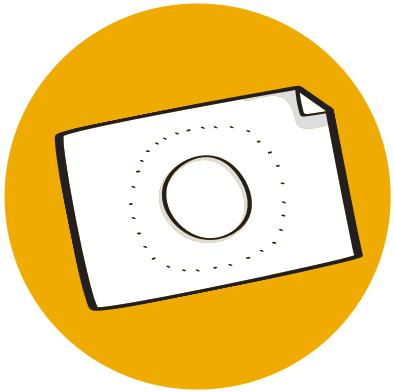
Define the scope of your architectural work

Statement of Architecture Work

Identify the scope and the business reason for your architecture

Understand context

What is the (business) reason of your project? How does it support the company's goals? Understand the architecture request and business background.



Define scope

What is your project about? Describe the scope including a high-level vision of the target architecture.

Acceptance Criteria & Roles

Think about acceptance criteria of your project and ways to check them. Who is participating in the project with which responsibility?

Statement of Architecture Work

Instructions

Duration

approx. 60 minutes

Input

- **Strategy Map**
- **Request for Architecture Work (Use-Case)**
- **Stakeholder Matrix**

Why & What

The Statement of Architecture Work defines the scope and approach that will be used to complete an architecture development. The Statement of Architecture Work is typically the document against which successful execution of the architecture project will be measured.

You can consider the Statement of Architecture work as a “Document of Understanding”. It helps to regain focus on the scope and avoids *scope creeping*. As it is formally agreed upon, the Statement of Architecture Work helps you to officially say NO to requests or requirements coming up throughout the architecture development that are out of scope.

How to use it

1. Define the title of your project and briefly summarize the reason for the architectural work including the business background. Use the Strategy Map as input.

2. Describe the architecture project with a focus on what is in scope of your architecture. One or two sentences with a clear description can be enough.

3. Visualize the project description and scope, with a simple high-level sketch of your architecture vision.

4. Add project roles and their responsibilities throughout the architecture development. Use the stakeholder matrix as input.

5. Define which stakeholders are responsible for accepting and approving your architecture. Use the Stakeholder Matrix as input.

6. Define key milestones of the architecture development.

Tips & Tricks

You can consider the Statement of Architecture work as a direct reaction to the identified use-case, i.e., the request of architecture work. You take insights from the Strategy Map as input for writing the Statement of Architecture Work.

Involve your key stakeholders.

Aligning on the scope of work, roles and responsibilities, and the necessary approvers will set expectations on resource needs and project timelines for the project.

Looking at the template for the Statement of Architecture Work, it can be divided into two parts:
(1) Rows 1 to 4 describes the aspired solution or use-case that was previously identified and articulated via the request of architecture work.
(2) Rows 5 to 7 have project management characteristics.

Statement of Architecture Work

Template

1. Title	*<Title of the project>*
2. Architecture project request and background	*<Short description of the reason for the project and its background>*
3. Architecture project description and scope	*<Brief description of the project and its scope>*
4. Overview of architecture vision	*<Provide a high-level picture of the target architecture, including core functional components and users / roles>*
5. Roles, responsibilities, and deliverables	*<List all the roles and their responsibilities in the project>*
6. Acceptance criteria and procedures	*<Describe the acceptance criteria and acceptance procedures of the project>*
7. Architecture project plan and schedule	*<Provide the main milestones and their schedule for delivering the project>*

Source: TOGAF Standard, Version 9.2

Statement of Architecture Work

Example

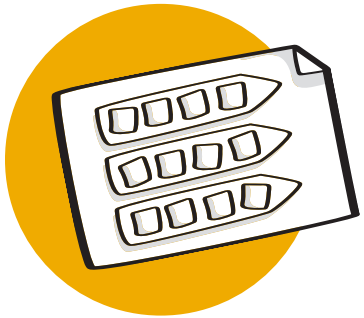
1. Title	Budget Request & Approval Solution
2. Architecture project request and background	To support the Global Growth Initiative 2025, design, develop, and operate a solution to support the financing of more than 2000 employees working on business dev. and R&D projects worldwide.
3. Architecture project description and scope	Plan for the development of a finance request & approval workflow aplication integrating with the existing finance system.
4. Overview of architecture vision (high-level-architecture)	
5. Roles, responsibilities, and work products	▪ **Business Architecture**: Director Business Development (Rocket Chips), Financial Analyst (Rocket Chips), Lead Architect (you) ▪ **Technical Architecture**: Lead Architect (you), Technical Architect (Vendor), Lead Developer (Rocket Chips), Head of IT (Rocket Chips) ▪ **Work products**: All artifacts from Lean EA toolkit without Architecture Principles, Risk Analysis and Use-Case Blueprint Diagram. Delivery format and content via PowerPoint and Word.
6. Acceptance criteria and procedures	▪ Approval for business architecture: Director Business Dev. (Rocket Chips), CFO (Rocket Chips) ▪ Approval for technical architecture: Head of IT (Rocket Chips)
7. Architecture project plan & schedule	▪ Architecture Roadmap work product, proposal for license and implementation effort

Discover Templates

Architecture

Principles

Instructions | Template | Example



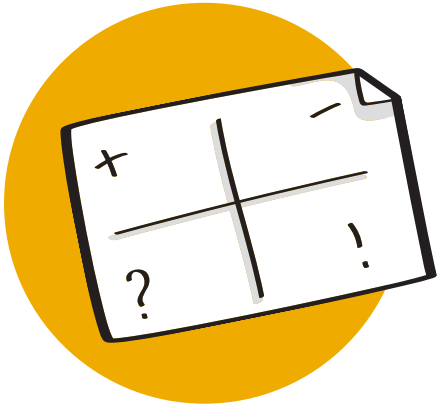
Constraints and guidelines that need to be considered for developing the architecture.

Architecture Principles

Guidelines you need to consider for developing the architecture

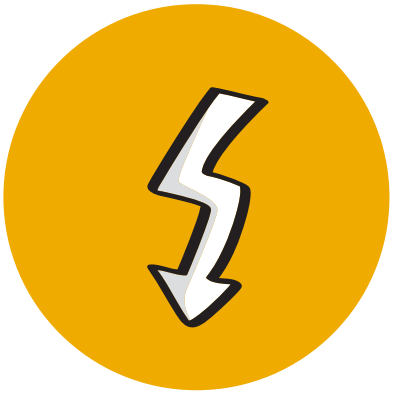
Identify principles

Do you need to consider specific standards, vendors, deployment options due to corporate strategy and market requirements?



Understand Benefits

Outline the business benefits adhering to the principle.



Identify impact

Identify the requirements for your architecture resulting from the architecture principle.

Architecture Principles

Instructions

Duration

approx. 30-60 minutes

Input

- **Statement of Architecture Work**
- **Corporate Guidelines**

Why & What

Architecture Principles define the constraints your architecture must deal with. Their purpose is to serve as a guideline you need to consider when developing the architecture.

These rules or constraints being described by the Architecture Principles have been defined in the past and are valid for the entire company. They define cooperate standards and strategic decisions that should be followed for the sake of TCO, operational efficiency and compliance, for example.

How to use it

1. Identify constraints you need to take into consideration while developing the architecture.

2. Understand the business benefits when following the principles.

3. Outline the impact of following the principles to your architecture development.

Tips & Tricks

As you do not want to simply copy and paste all principles that have been defined, identify those principles that are relevant for your architectural work. For this evaluation, you take the Statement of Architecture Work, and the scope being defined there, into consideration.

Architecture Principles
Template

Name	*Represents the essence of the rule. Easy to remember.* *Specific technology platforms should not be mentioned in the name or statement of a principle.*	**# ID**
Statement	*Unambiguously communicate the fundamental rule.*	
Rationale	*Highlight the business benefits of adhering to the principle using business terminology. Describe the relationship to other principles and the intentions regarding a balanced interpretation.*	
Implications	*Highlight the requirements, both for the business and IT, for carrying out the principle in terms of resources, costs, and activities. It will often be apparent that current systems, standards, or practices would be incongruent with the principle upon adoption. The impact to the business and consequences of adopting a principle should be clearly stated.*	

Source: TOGAF Standard, Version 9.2

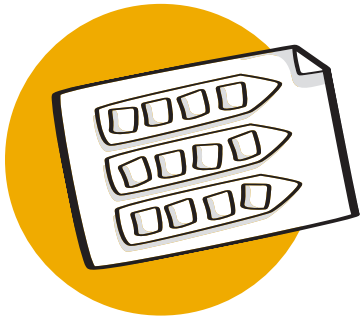
Architecture Principles
Example

Name	**Preferred IT vendor strategy**	**BP_020**
Statement	Consider applications from Rocket Chips strategic IT partners first: Microsoft and SAP	
Rationale	Rocket Chips has long relationships to its IT partners (vendors and services) which are based on corporate contracts to ensure best license prices, interoperability and integration, maintenance and premium support (e.g., 24x7).	
Implications	• Organizations may not be able to select the best fit-for-purpose application from an ISV when our partners offer similar capabilities. Although, the individual cost may be competitive, the overall corporate expenditures are easier to control through our partner contracts. • Maintenance and support contracts are already in place through corporate contracts. • International availability can be ensured.	

Source: TOGAF Standard, Version 9.2

Risk
Analysis

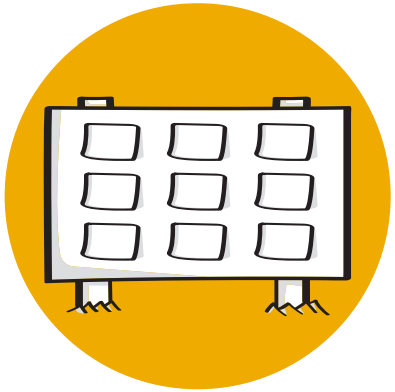
Instructions | Template | Example



Identify, classify, and mitigate risks for the architecture

Risk Analysis

Identify, classify, and mitigate risks for the architecture



Identify Risks

Identify and classify initial risk with respect to impact to the organization/ business unit.

Plan Mitigation

Identify and plan mitigation actions. Re-asses the risk to classify the residual risk level.

Manage Risks

Execute risk mitigation and monitor execution.

Risk Analysis

Instructions

Duration

Ongoing

Input

- **Strategy Map**
- **Stakeholder Matrix**
- **Statement of Architecture Work**

Why & What

There is always risk associated to the architecture you're developing: risks that the architecture will fail, i.e., it cannot be developed, it cannot be operated, or is not in line with other ongoing projects.

How to use it

1. Identify and briefly describe the risks. Classify them according to the categories high, medium and low (initial level of risk). Describe the impact of the initial level of risk to the architecture.

2. Define actions for mitigating the risks identified. Actions can range from an additional level of stakeholder management, to identifying reference architectures solving a similar request for architectural work.

3. Re-assess the risk level and assign the residual level of risk. Describe the impact of the residual level of risk to the architecture.

Tips & Tricks

When you think about risk, you can distinguish between two levels of risk: the *initial level of risk* and the *residual level of risk**.

You can think of three categories for risks*:

- (H)igh Risk: Significant failure of parts of the architecture project. Certain goals of the organization/ business unit will not be achieved.
- (M)oderate Risk: Noticeable failure of parts of the architecture project threatening the success of certain goals of the organization/ business unit.
- (L)ow Risk: Certain goals of the organization/ business unit will not be fully successful.

Always remain in the scope of your architecture as defined in the Statement of Architecture Work.

You can decide to only define mitigation actions for risks having an initial level of high and medium.

* Source: TOGAF Standard, Version 9.2

Risk Analysis
Template

Risk ID	Risk	Initial Risk		Mitigation	Residual Risk	
		Classification	Impact		Classification	Impact
<id>	*<risk description>*	*<initial risk classification>*	*<description of impact of initial risk on the organization or architecture>*	*<description of mitigation action>*	*<risk level after mitigation action>*	*<description of impact mitigation action and of residual risk on the organization or architecture>*

Source: TOGAF Standard, Version 9.2

Risk Analysis
Example

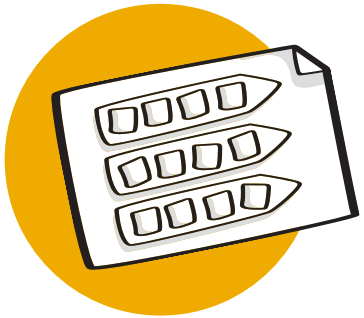
Risk ID	Risk	Initial Risk		Mitigation	Residual Risk	
		Classification	Impact		Classification	Impact
R_001	*Low user acceptance*	*High*	*Anticipated business improvements are not realized. Growth Strategy is not executed properly*	*Closely include the users and business stakeholders right from the start. Create clickable low-fidelity prototypes and understand the user journey by applying design thinking tools. Provide user training early and conduct regular education sessions.*	*Low*	*Additional effort. There might still be several users who do not accept the solution.*

Source: TOGAF Standard, Version 9.2

Solution

Context Diagram

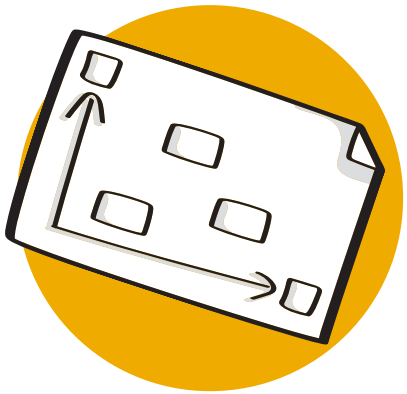
Instructions | Template | Example



Shows the relationship between the proposed solution and the organizational units, business roles, and business functions within the enterprise.

Solution Context Diagram

Shows the relationship of the aspired solution to the organization

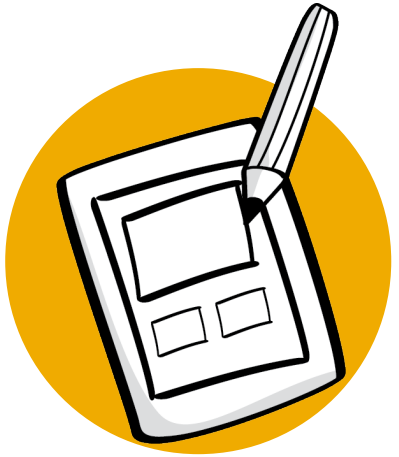


Identify business capabilities

Identify the key objectives, main use-case(s) and business capabilities of the aspired solution.

Capture Business Users

Identify related organizational units, business roles, users and existing business applications.



Visualize solution context

Visualize the aspired solution as one component including relationships to organizational units, business functions and business solutions.

Solution Context

Instructions

Duration

approx. 30-60 minutes

Input

- **Statement of Architecture Work**
- **Stakeholder Matrix**

Why & What

The goal of the Solution Context is to provide a high-level overview of the aspired solution that can be easily understood by business. Therefore, the Solution Context describes the required *business capabilities* that need to be satisfied by the architecture.

Input for creating the Solution Context are the Statement of Architecture Work and the Stakeholder Matrix.

Specifically, for stakeholders from the business domain, the Solution Context is a good visual representation of the architecture, showing how your anticipated solution interacts with different organizational units, roles and business functions.

How to use it

1. Translate your learnings about the aspired solution into business capabilities, that are describing *what* the aspired solution can do. These can be main functions or features of the aspired solution expressed in business terminology. This list doesn't need to be exhaustive; think of 5 to 10 main capabilities.

2. Identify related organizational units, business roles, and existing business applications. For the identification of users and roles, use the Stakeholder Matrix as input.

3. Visualize the input in a diagram (refer to template).

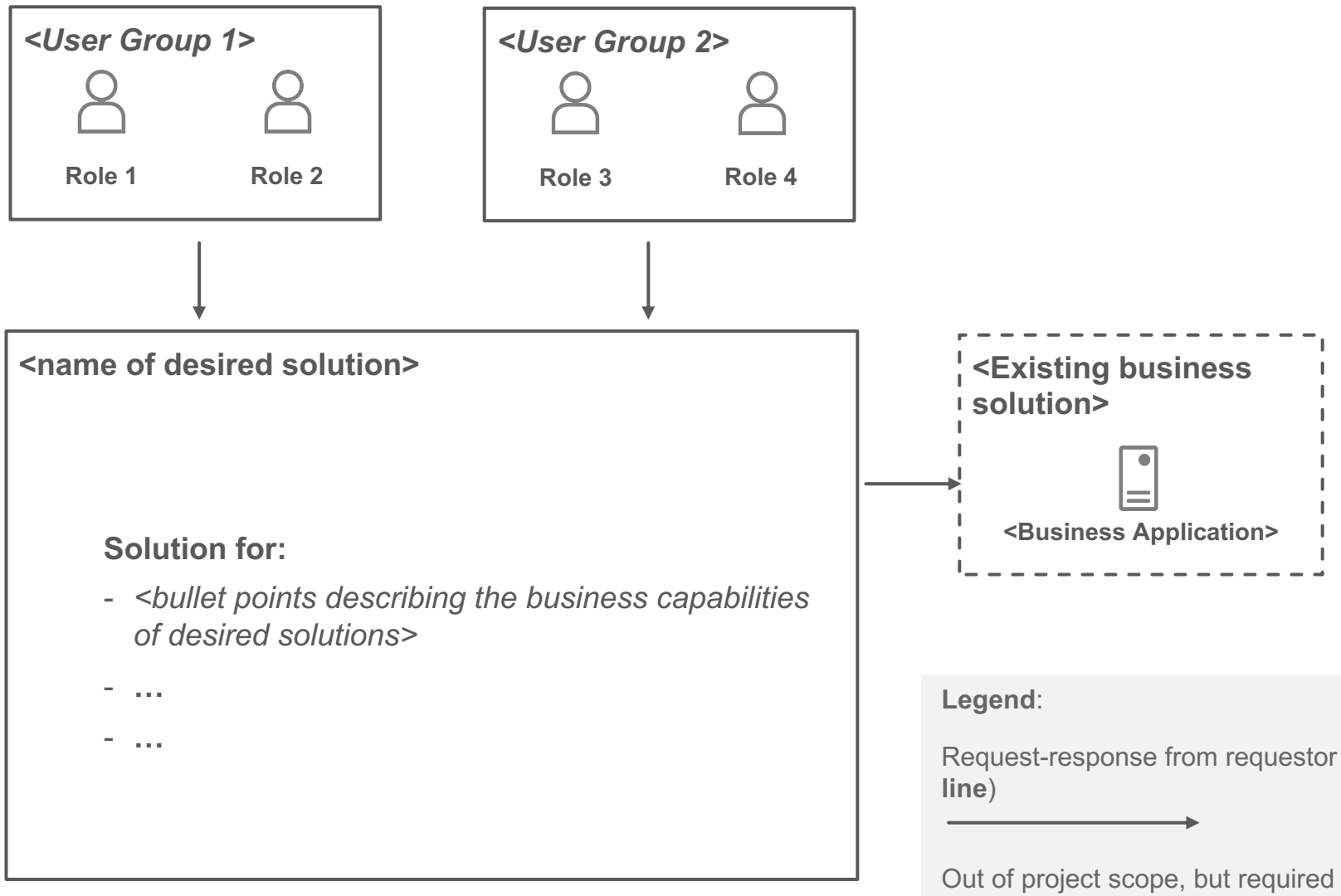
Tips & Tricks

The most prominent part of the Solution Context Diagram is "the box" representing the aspired solution. You label the box with the name you have defined in the Statement of Architecture Work.

Throughout architecture development, you gradually add more details to the Solution Context, i.e., evolving the Solution Context to other work products such as a Solution Concept and a Solution Realization Diagram.

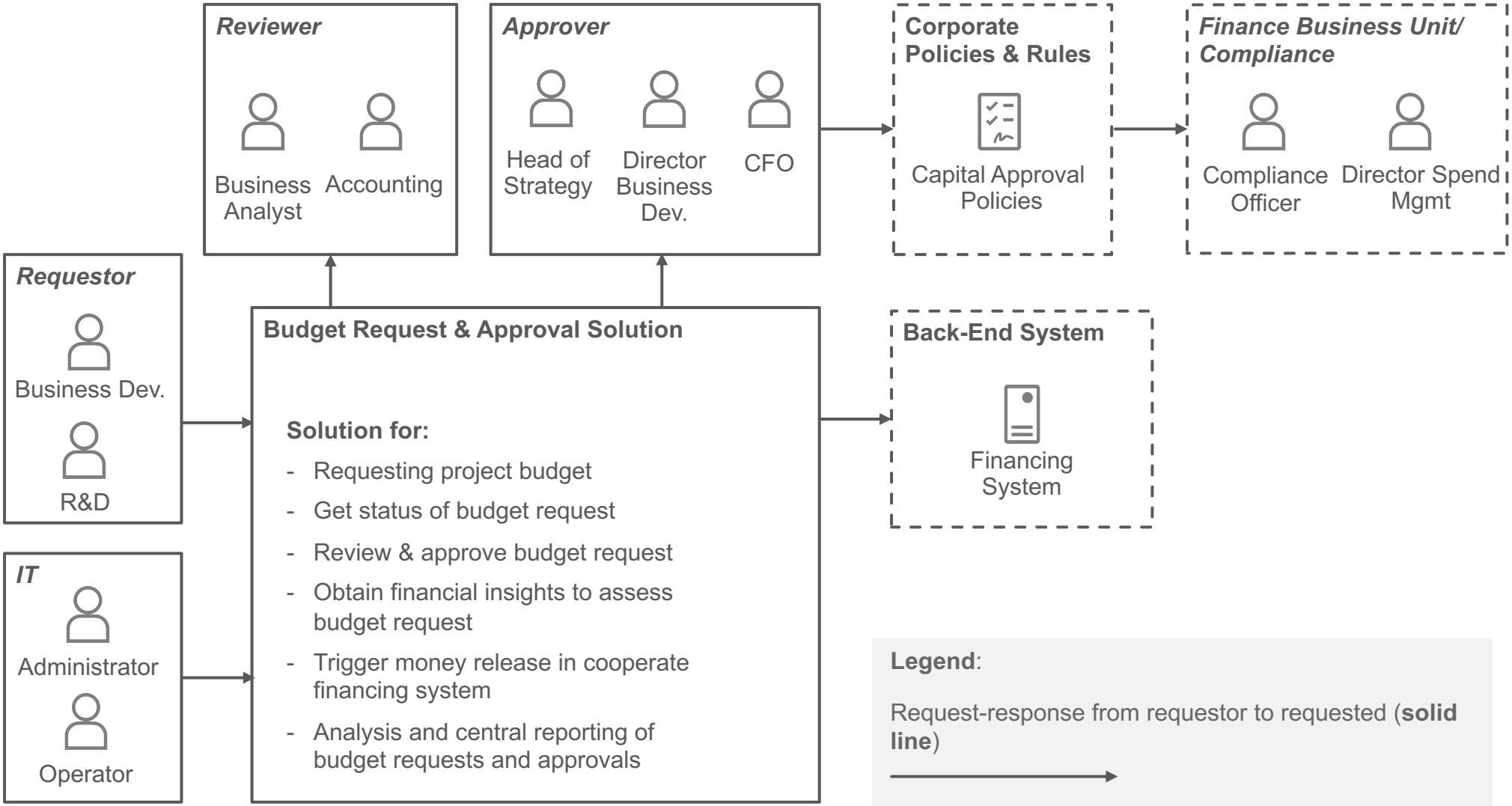
You use the Solution Context to share and communicate your architecture to the business stakeholders. You can also update your Statement of Architecture Work by adding the Solution Context to the "Overview of architecture vision" section of the template.

Solution Context Diagram
Template



Solution Context Diagram

Example

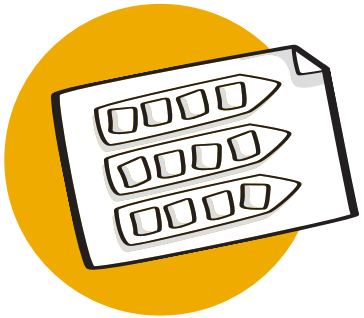


Design Templates

Use-Case

Blueprint Diagram

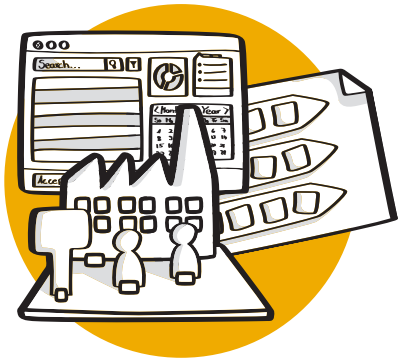
Instructions | Template | Example



Pivot from Design Thinking to Architectural Thinking by mapping user actions to architectural requirements such as applications, data and specific technical features.

Use-Case Blueprint Diagram

Pivot from Design Thinking to Architectural Thinking

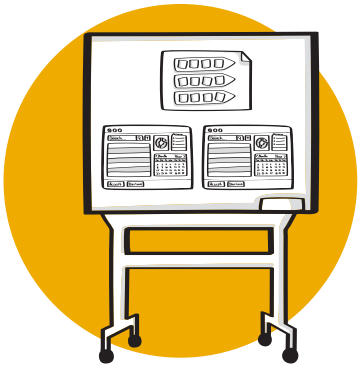


User Story

Visualize and describe a specific use-case/ scenario of a specific persona with a storyboard and corresponding user actions.

Data & Applications

Identify which data, what applications are required for each user-action along the storyboard.



Technical capabilities

Identify technical capabilities that are required for each user-action along the storyboard.

Use-Case Blueprint Diagram

Instructions

Duration

approx. 30-60 minutes

Input

- **Statement of Architecture Work**
- **Stakeholder Matrix**
- **Solution Context**

Why & What

The purpose of the Use-Case Blueprint Diagram is to pivot from Design Thinking to Architectural Thinking. User-centric actions are mapped to technical aspects of the architecture, such as data, systems, and technical capabilities.

The Use-Case Blueprint diagram, with its user centricity, is the bridge to creative thinking and the Design Thinking methodology. In case you want to add more user-centricity and usability focus to your architecture, you find additional work products in the Design Thinking Methodology (Innovation Toolkit by SAP AppHaus).

How to use it

1. Create a storyboard by putting yourself in the shoes of a key user. Describe the desired objective or capability by disassembling the objective into individual scenes or scenarios for the key user.

2. Think about technical requirements that need to be met in order to realize the user actions: What data is required for the actions in the scenes? Which applications and IT systems are involved in the user actions?

3. Based on your experience, add technical capabilities that are required at a specific step or action, like IoT data ingestion, a chatbot or a workflow service, for example.

Tips & Tricks

Taking the previously identified main-uses and key-objectives of the users from the Solution Context as input, put yourself in the shoes of one or several key-users. Create one Use-Case Blueprint Diagram per user and per key objective of this user. As this can clearly be a lot of work, you might want to choose the most valuable objective and user combination and will not do all possible combinations.

You will re-use information form the Use-Case Blueprint Diagram at later steps in the technical architecture domain. For example, the data identified, helps to understand the information flow, which is represented in the Solution Concept and the Solution Realization Diagram. Also, the identified data serves as input for the Conceptual Data Diagram.

The identified applications are used for the creation of Baseline Solution Architecture, the Solution Concept and Solution Realization Diagram.

The identified technical capabilities support the choice of Solution Building Blocks.

Use-Case Blueprint Diagram

Template & Example

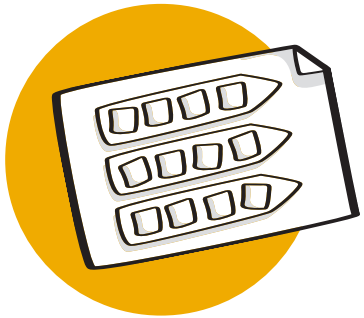
	Template:	**Scene 01**: Business developer wants to request budget for new project idea.	**Scene 02:** …
Target Scenario How does the user work with the target solution?	*Describe the context / scene of the user*		
Actions What actions does the user perform to achieve desired results?	*Describe what the user is doing*	▪ Open “Request Application” ▪ Enter project details ▪ …	▪ …
Required Data What data is required for the action?	*Describe the data, its attributes, and data type*	▪ Project name + description ▪ Project goals + KPIs ▪ …	▪ …
Systems & Applications Which technical systems or applications are accessed for the action?	*List systems and applications required to perform the action*	▪ “Request Application” ▪ “Central Repository” ▪ …	▪ …
Technical Cabability Which technical capabilties are required to perform the action?	*Highlight required technical capabilities to perform the action*	▪ Check integrity of project details (e.g. time vs. budget, …) ▪ Automatically consider budget limits ▪ …	▪ …

Technology Architecture Domain

Baseline

Solution Architecture

Instructions | Template | Example



Describe existing applications and IT components relevant for the use-case (request of architecture work)

Baseline Solution Architecture

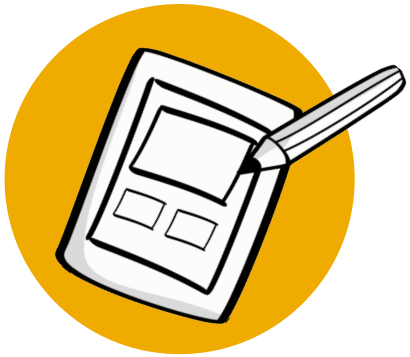
Describe existing components relevant for your architecture

List current applications

Identify existing applications and IT components that are relevant for the request of architecture work (use-case).

Understand dependencies

Identify the functional relationship between the identified components.



Visualize with diagram

Draw the existing applications and IT components including their dependencies and relationships. Add users and roles.

Baseline Solution Architecture

Instructions

Duration

approx. 30-60 minutes

Input

- **Stakeholder Matrix**
- **Statement of Architecture Work**
- **Use-Case Blueprint Diagram**

Why & What

Describes applications, software components, and functional components currently in use and relevant for the use-case/ request of architectural work.

The reason for creating the Baseline Solution Architecture is to identify building blocks that can be re-used for designing the architecture or identify components which need to be integrated via interfaces like APIs, Events or Data replication, for example.

It also shows the scope of change initiatives resulting from the aspired solution to the current IT landscape.

How to use it

1. List all the current applications and IT components (Building Blocks) that are relevant for the use-case (scenario).

2. Understand the dependencies between the identified IT components (functional dependency in terms of request-response and/ or information flow)

3. Add users and roles currently interacting with the IT components. It is likely, that some users and roles are identical to the ones identified for the Solution Context Diagram and are also part of the Stakeholder Matrix.

Tips & Tricks

The Baseline Solution Architecture is important, because it shows what IT components, or Building Blocks, are already in place and can be considered when creating the Solution Concept and Solution Realization Diagram in the next steps.

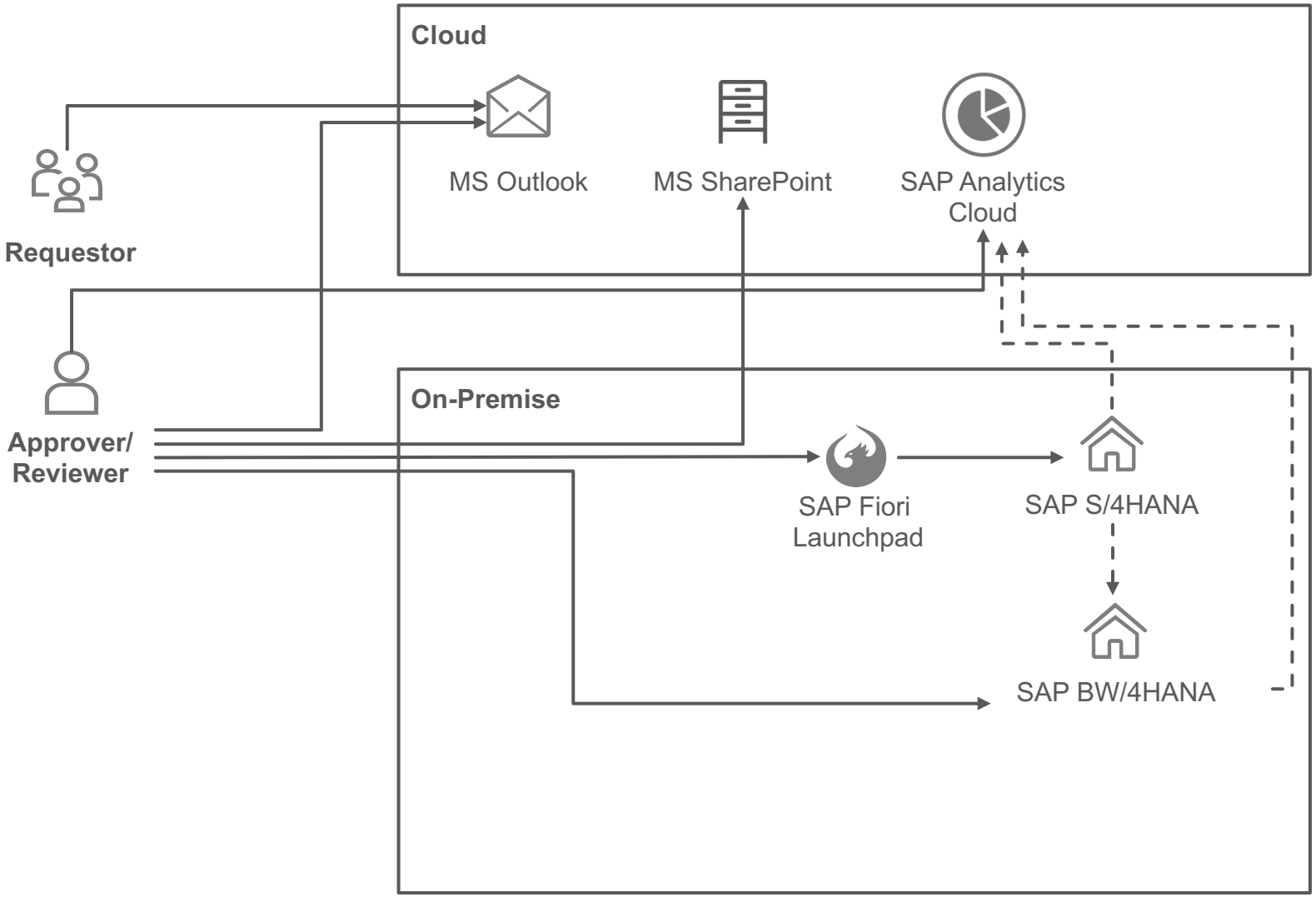
Considering the Statement of Architecture Work, it is worth emphasizing that not the complete corporate IT landscape needs to be represented in the diagram. Instead, focus on the areas of interest, the areas that are in scope of your architecture engagement/ request of architecture work.

The Use-Case Blueprint Diagram can be used to identify existing IT systems relevant for the architecture.

Consult IT and business stakeholders knowing the context of your use-case (the request of architecture work) to understand which existing IT components should be considered.

Baseline Solution Architecture

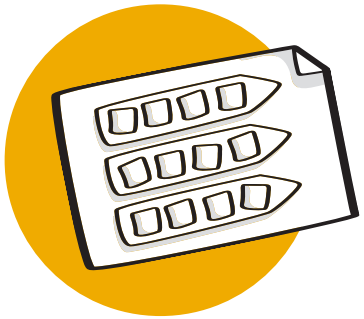
Example



Solution

Concept Diagram

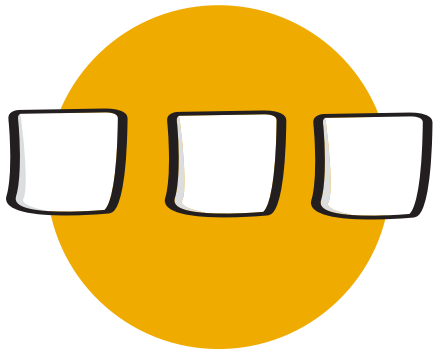
Instructions | Template | Example



A a high-level representation of the aspired solution via Architecture Building Blocks

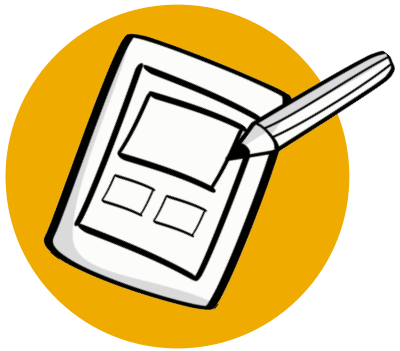
Solution Concept Diagram

A high-level representation of the aspired solution via Architecture Building Blocks (ABBs)



Identify architecture building blocks

Identify high-level building blocks to satisfy the key objectives of the aspired solution. Map all business capabilities from the Solution Context Diagram to one or more ABB(s).



Draw Diagram

Draw the architecture building blocks including their relationships and dependencies.

Add Business Units

Add organizational units, functions and users or roles including relationships to the architecture building blocks.

Solution Concept Diagram

Instructions

Duration

approx. 30-60 minutes

Input

- **Solution Context Diagram**
- **Baseline Solution Architecture**

Why & What

Provides a high-level representation of your solution that is envisioned in order to meet the requirements of your architecture engagement. You can consider the solution concept as a “pencil sketch” of your expected solution based in architecture building blocks (ABBs).

The purpose of the Solution Concept is to provide a high-level representation of your solution. You can think of the Solution Concept as a first pencil-sketch showing Architecture Building Blocks of your solution.

How to use it

1. Identify high-level architectural building blocks that are needed to satisfy the key objectives and business capabilities as described in the Solution Context Diagram. Translate the business-oriented capabilities describing the aspired solution in the Solution Context, to respective Architecture Building Blocks.

2. Outline the relationships between the identified building blocks. Relationships can be of type request-response or information flow.

3. Add users, roles, and organizational units including their relationship to the architecture building blocks.

Tips & Tricks

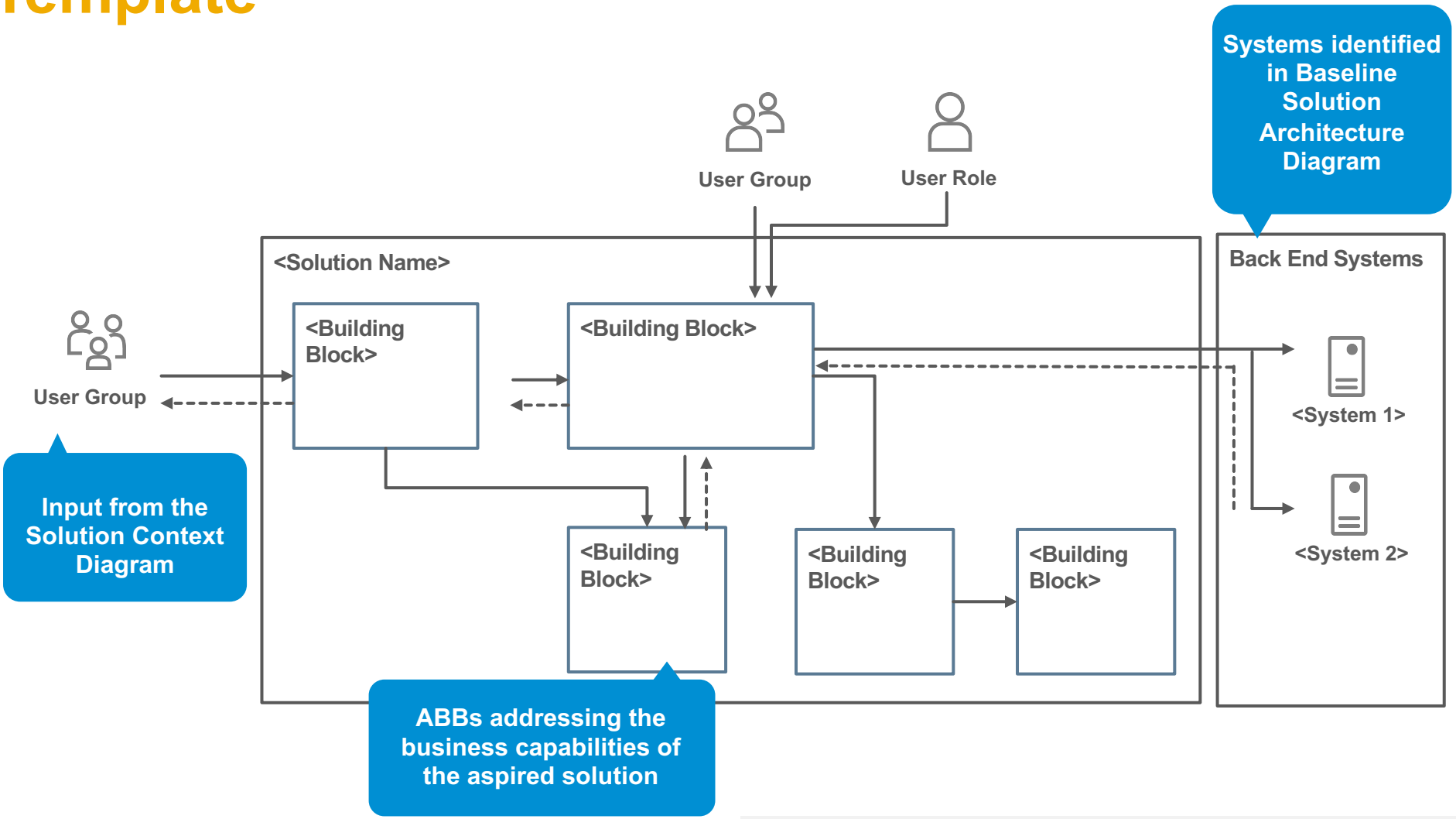
The Solution Concept is a transition step between the Solution Context and the Solution Realization Diagram. It is a macro version of your architecture, mainly comprised of vendor neutral Architecture Building Blocks.

By mapping the capabilities of the aspired solution to architecture building blocks, it might be the case that one business capability directly translates to one ABB. Also, it can be the case that one business functionality translates to many ABBs related to each other.

A good quality check of the Solution Concept Diagram is to check if all required business capabilities are addressed with corresponding ABBs.

Solution Concept Diagram

Template



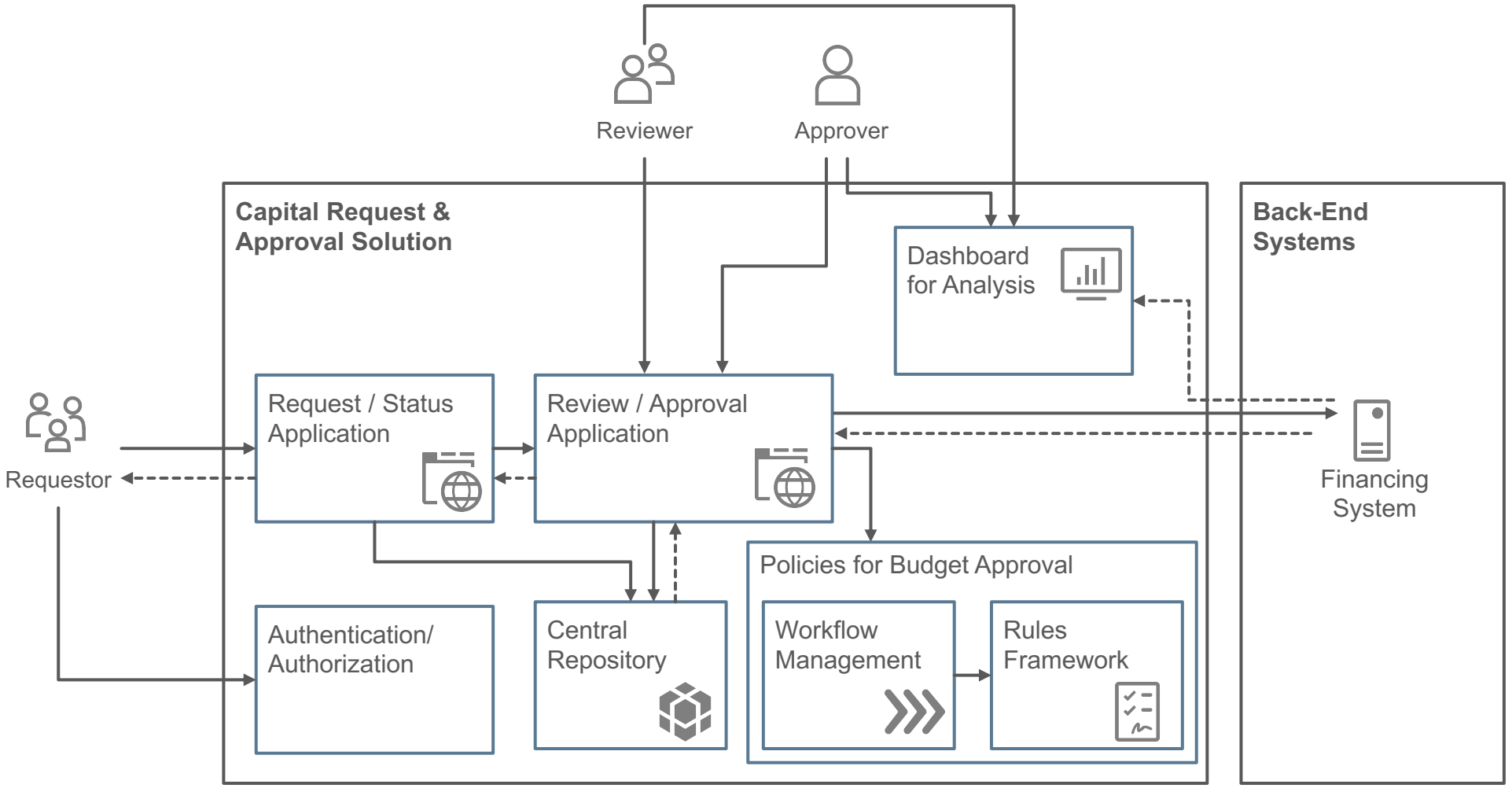
Legend:

Request response:

Information flow from data source to target:

Solution Concept Diagram

Example



Legend:

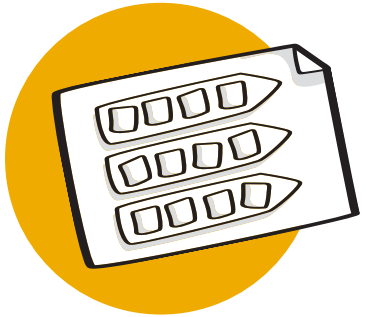
Request response: ——→

Information flow from source to target: ←- - - -

Solution

Realization Diagram

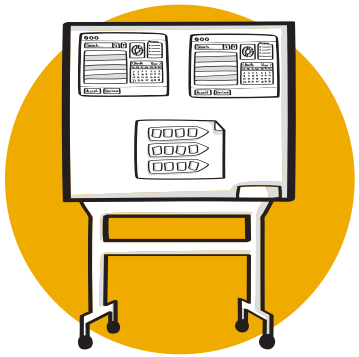
Instructions | Template | Example



A technical description of the aspired solution via Solution Building Blocks

Solution Realization Diagram

A technical description of the aspired solution via Solution Building Blocks

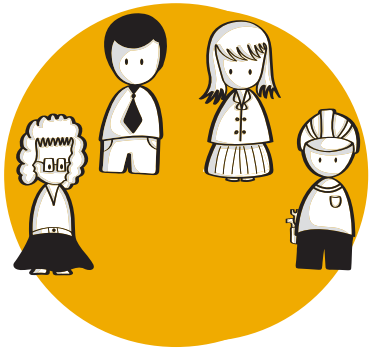


Identify Solution Building Blocks

Perform a Fit-Gap Analysis of the Baseline Solution Architecture. Map vendor specific solution building blocks* to architecture building blocks.

Identify functional dependencies

Identify dependencies and the data flow between the solution building blocks.



Add users

Add users & roles including their relationship to the solution building blocks.

* based on experience, reference architectures, solution descriptions.

Solution Realization Diagram

Instructions

Duration

approx. 60-120 minutes

Input

- **Baseline Solution Architecture**
- **Solution Concept Diagram**
- **SAP Discovery Center (vendor specific information sources)**
- **SAP API Business Hub**

Why & What

Describes your target architecture by breaking the aspired solution into functional components based on solution building blocks (SBB). The purpose of the Solution Realization Diagram is to add technical details so that it can be used for the implementation phase of the aspired solution.

Solution Building Blocks are product or vendor-aware and can be either developed or procured. With the Solution Building Blocks you outline which products, like cloud services, are used, or need to be developed, in order to realize and implement the aspired solution/ its architecture building blocks.

How to use it

1. Identify Solution Building Blocks that realize the previously identified Architecture Building Blocks. Map the Architecture Building Blocks of the Solution Concept to one ore more Solution Building Blocks. Sources for Solution Building Blocks are the Baseline Solution Architecture (perform a fit-gap-analysis) and vendor specific information sources like the SAP Discovery Center for SAP BTP Solution Building Blocks.

2. Outline the dependencies between the identified SBBs (functional dependency in terms of request-response and/ or information flow)

3. Add users and roles currently interacting with the SBBs.

4. Check the Solution Realization Diagram for completeness, by verifying that each ABB from the Solution Concept is addressed with SBBs.

Tips & Tricks

There are basically two sources of Solution Building Blocks:

(1) Baseline Solution Architecture. Based on the Baseline Solution Architecture you perform a Fit-Gap-Analysis, identifying Solution Building Blocks that are already in place and can either be completely or partly re-used in your Solution Realization.

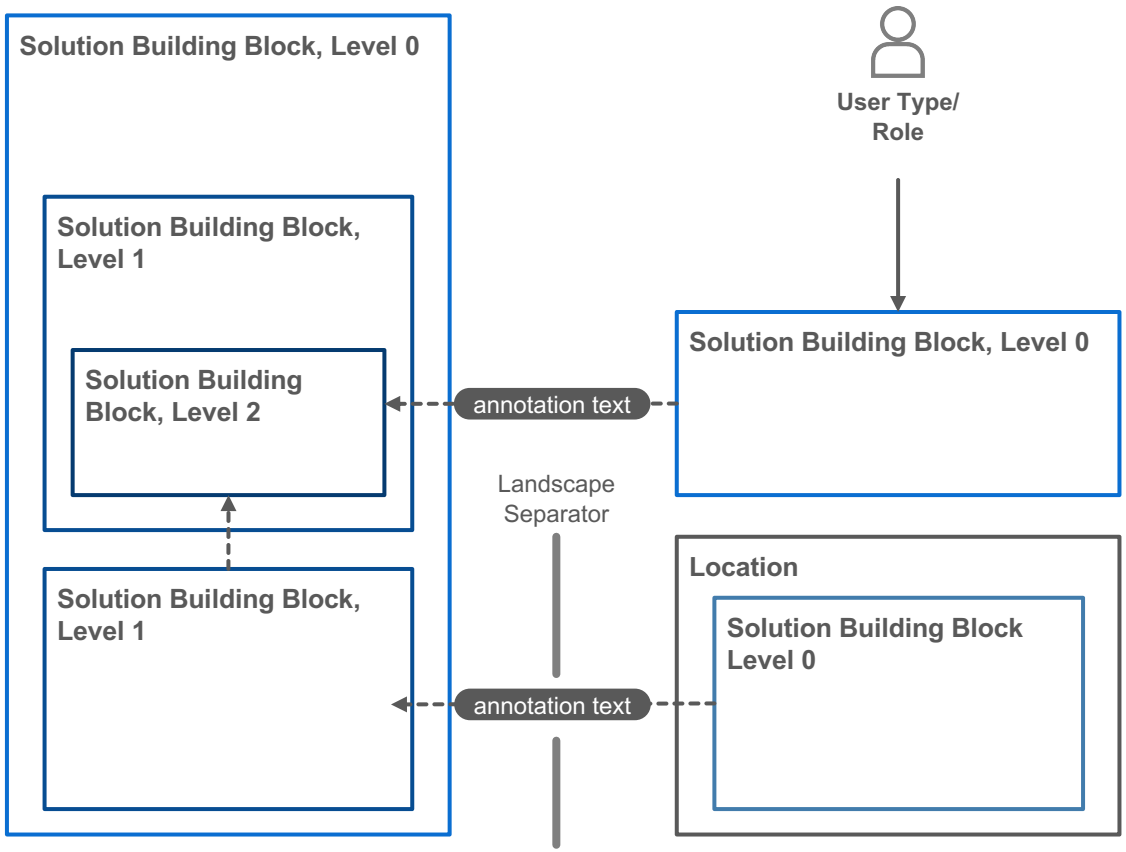
(1) SAP Business Technology Platform services (e.g., via SAP Discovery Center).

The mapping of Solution Building Blocks to Architecture Building Blocks depends on the experience and knowledge in the team (experience-based mapping).

Publicly available information/ documentation can be consulted about the Solution Building Blocks for doing the mapping. Reference architectures solving a similar problem or addressing parts of your request of architecture work are also good sources for doing the mapping.

Use BTP service icons to draw the diagram.

Solution Realization Diagram
Template



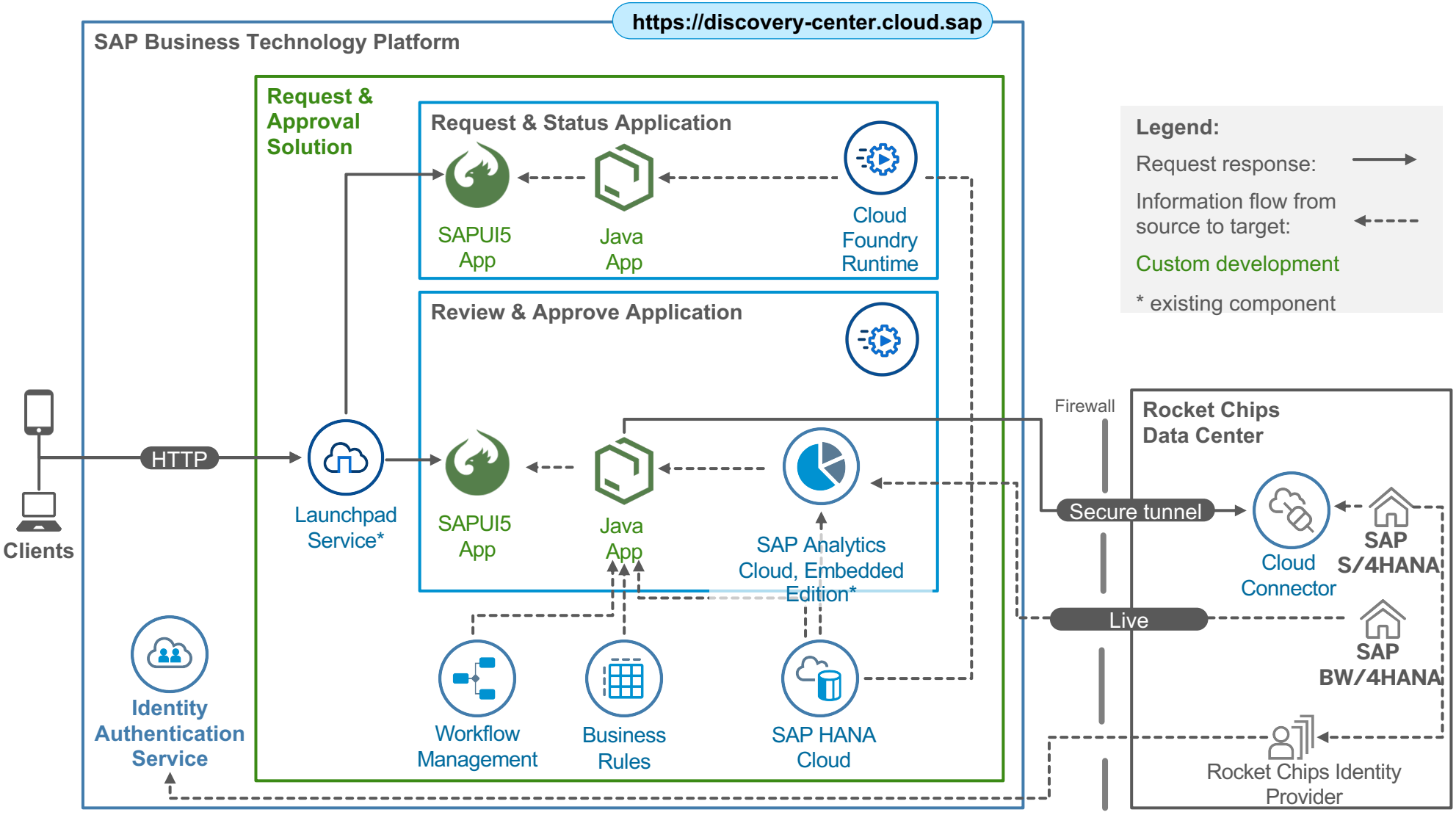
Legend:

Request response: ⟶

Information flow from source to target: ⟵ - - - -

Solution Realization Diagram

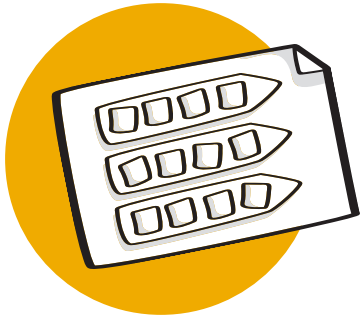
Example



Architectural
Decisions

Instructions | Template | Example

Document your architectural decisions and reasoning



Architectural Decisions

Document your architectural decisions and reasoning

Keep track of alternatives

Identify discussions for different architectural alternatives. What alternatives do you consider?

Document your thoughts

Document the reasoning for the decision taken. Why did you make this decision?

Keep updating

Continuously update your architectural decisions throughout the architecture development process.

Architectural Decisions

Instructions

Duration

ongoing

Input

- **Solution Concept Diagram**
- **Solution Realization Diagram**
- **Software Distribution Diagram**
- **Environments & Location Diagram**

Why & What

Document the architectural decisions you have chosen for realizing the target architecture.

The purpose is to document your architectural discussions and outline the reason for the decisions taken.

How to use it

1. Every time you consider two or more alternatives in the architecture, you should remind yourself to write down the reasoning behind your decision for one of the alternatives.

2. Note the reason why you have chosen the alternative. Was it due to corporate strategy? Maybe there is a corresponding architecture principle that you considered during the decision? You also can include a SWOT analysis to explain the reasoning of your decision.

Tips & Tricks

Throughout the creation of your architecture, you make different decisions. For example, you decide to use one solution building block of vendor A instead of another alternative from vendor B.

Or you decide to implement a required solution building block by your own, instead of purchasing a building block.

All these decisions are documented for later reference and show the reasoning behind your decision.

Documenting your architectural decisions is a recurring task and it is likely that you add decisions at later stages of your architecture development. For example, when you think about the deployment of your architecture in the context of the environments and location diagram.

Architectural Decisions

Template & Example

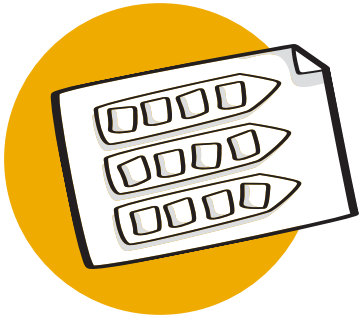
Ref	Decision	Reasoning
<unique identifier>	*<short description of the decision>*	*<details about the reason for the decision and possible alternatives that were considered>*
An_100 September-2020	Provide embedded analytics to support profitability analysis of budget requests.	To offer an integrated user experience and integrated end-to-end experience, SAP Analytics Cloud, embedded edition is used and integrated into the “Budget Request & Review application”.
…	…	…

Conceptual

Data Diagram

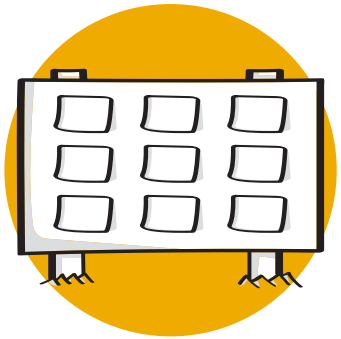
Instructions | Template | Example

Describes information objects processed by the architecture



Conceptual Data Diagram

Describes information objects processed by the architecture.

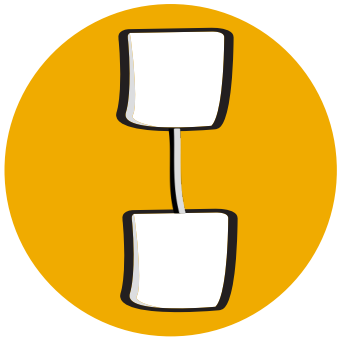


Identify entities

Identify (business) entities the architecture needs to process.

Define Attributes

Add attributes (properties) and data types to the entities.



Define Relationships

Identify and draw relationships between the entities.

Conceptual Data Diagram

Instructions

Duration

approx. 60-120 minutes

Input

- **Use-Case Blueprint Diagram**
- **Solution Realization Diagram**

Why & What

Describes entities (information objects) processed by the architecture.

The purpose of the conceptual data diagram is to outline the relationship between data and business entities of the aspired solution.

How to use it

1. Identify information objects (entities) being processed by the Solution Building Blocks in the Solution Realization Diagram and understand the information flow between the SBBs.

2. Name the entities you have identified with an unambiguous name and add attributes or properties with respective data types to further describe the entities. For choosing data types, keep it simple and chose types like “number”, “string”, “date”, for example.

3. Think about the relationship between the different entities. Mostly, you will design an association between two entities and define a multiplicity. The association is represented by a solid line between two entities and is described with a verb.

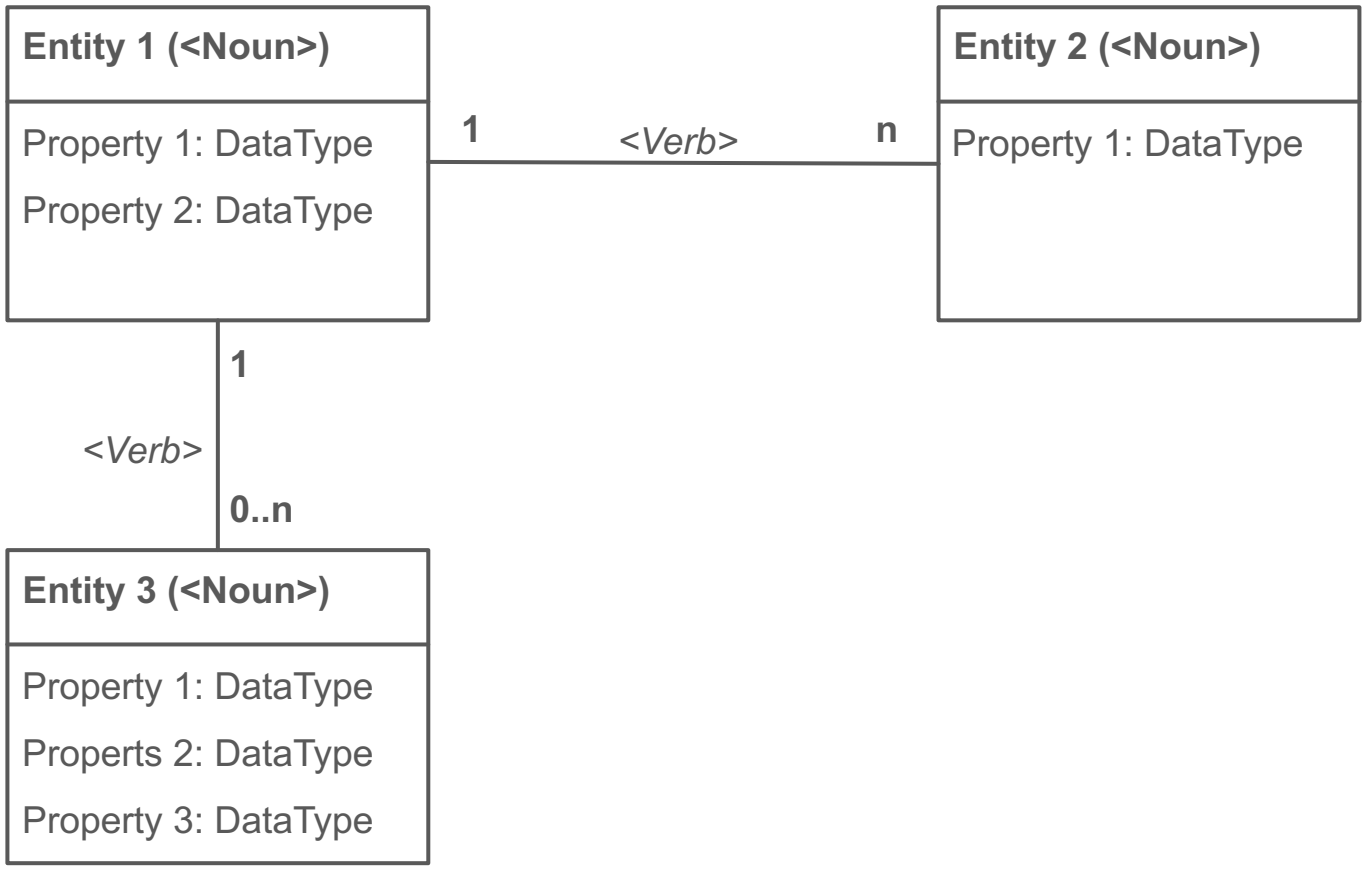
Tips & Tricks

The Conceptional Data Diagram can be treated like an Entity-Relationship Diagram, outlining, and describing the information being processed by the aspired solution.

However, the conceptual data diagram is not a technically detailed data model description which you can directly map to a relational data model, for example. In fact, the information objects, or data entities, you are describing via the Conceptional Data Diagram can be stored and handled by different solution building blocks of the architecture. They might not necessarily end up in a relational database, maybe a graph store is suited better, or you get the information objects via an API or message queue, from a building block outside the scope of your architecture.

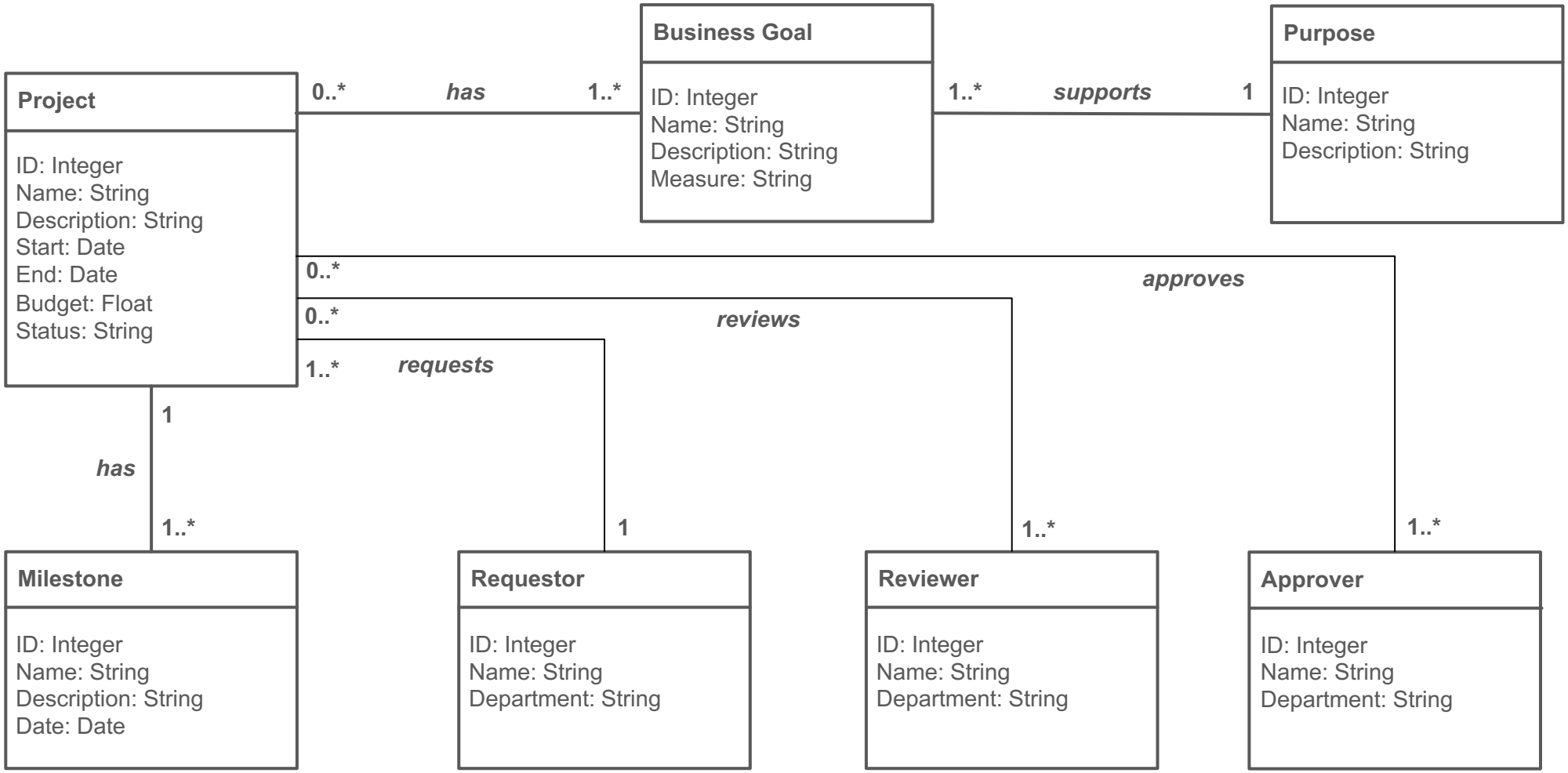
One source of input for creating the Conceptual Data Diagram is the Use-Case Blueprint Diagram. Another source of input for creating the diagram is the Solution Realization Diagram.

Conceptual Data Diagram
Template

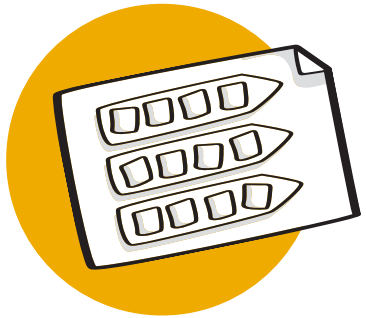


Conceptual Data Diagram

Example



Software
Distribution Diagram

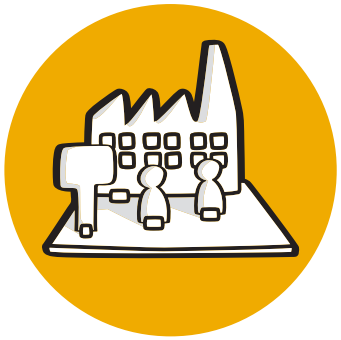


Instructions | Template | Example

Shows how architecture- and solution building blocks are distributed across the IT infrastructure

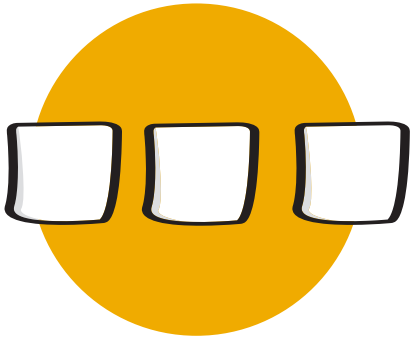
Software Distribution Diagram

Shows how architecture- and solution building blocks are distributed across the IT infrastructure.



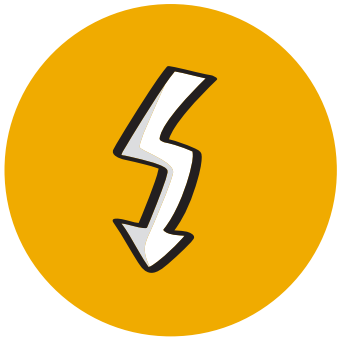
Deployment Environments

Identify relevant hosting and deployment environments of your solution (e.g., cloud and on-premise).



Map Building Blocks

Map the building blocks of your architecture to the hosting and deployment environments.



Data Flow

Visualize the data flow/ response-request relationship between building blocks indicating network communication between different environments.

Software Distribution Diagram

Instructions

Duration

approx. 30-60 minutes

Input

- **Baseline Solution Architecture**
- **Solution Concept Diagram**
- **Solution Realization Diagram**

Why & What

Shows how the aspired solution is structured and how the solution building blocks / architecture building blocks are distributed across the IT infrastructure.

The purpose of the Software Distribution Diagram is to enable a view of how the aspired solution is deployed and hosted.

How to use it

1. Identify relevant hosting and deployment environments for the building blocks of the architecture. A classification between cloud and on-premise might be sufficient.

2. Map the building blocks of the architecture to the identified hosting and deployment environments.

3. Visualize the relationships of type request-response or information flow between the building blocks. This helps to understand potential latency considerations or network requirements .

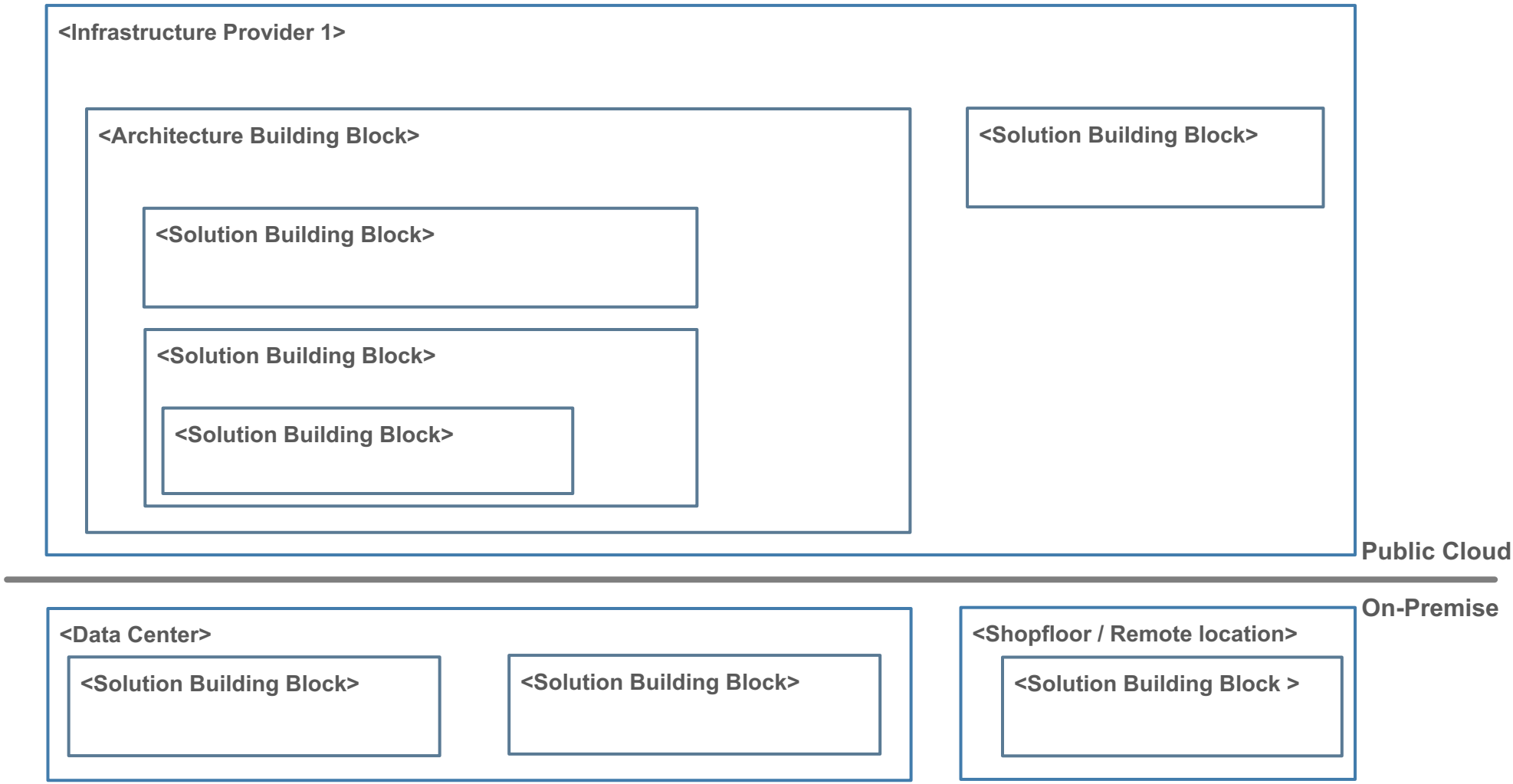
Tips & Tricks

As a corporate IT infrastructure has typically grown beyond the four walls of a corporate data center, the purpose of the software distribution diagram is to show in which “landing zones” the different building blocks of the architecture are running. As today’s IT infrastructures are typically hybrid, you can use a general classification in On-Premise- and Cloud- landing zones. The On-Premise landing zone can be further categorized into specific data center, shopfloors or stores, for example. The cloud landing zone can be further categorized into IaaS provider or SaaS provider, for example.

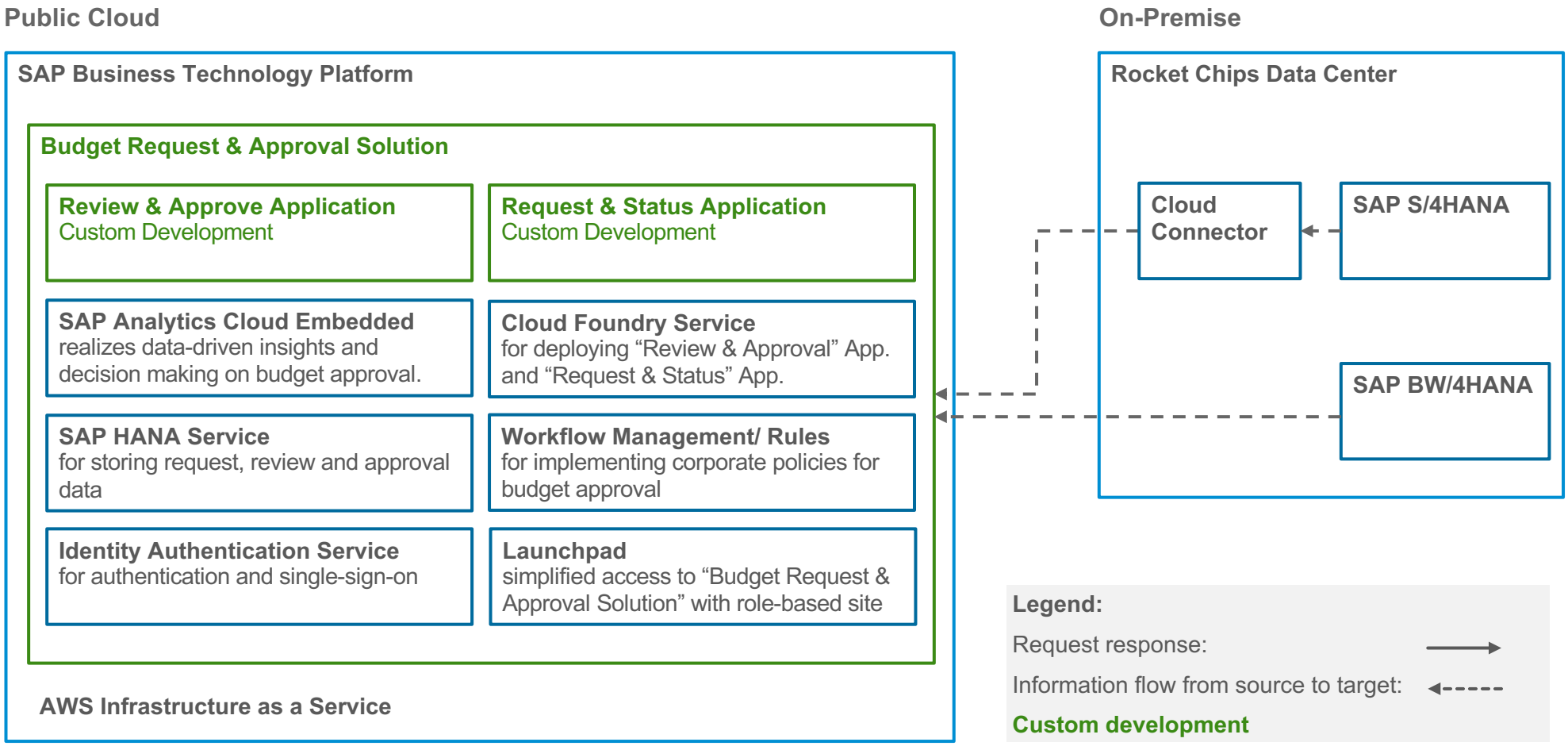
For creating the Software Distribution Diagram, you need the list of building blocks making up your solution. Consult the Solution Concept Diagram to get the list of architecture building blocks and the Solution Realization Diagram, to get the list of solution building blocks.

Software Distribution Diagram

Template



Software Distribution Diagram
Example

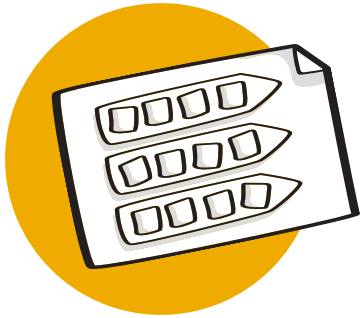


Deliver Templates

Environments &
Location Diagram

Instructions | Template | Example

Shows the geographical location of building blocks and runtime environment specific details



Environments & Location Diagram

Shows the geographical location of building blocks and runtime environment specific details

Deployment Environments

Identify deployment environments (i.e., infrastructure, runtime environments) that are running solution building blocks of your architecture.

Choose locations

Choose the geographical locations of the previously identified deployment environments.

Data Flow

Visualize the data flow/ response-request relationship between building blocks indicating network communication between different environments.

Environments & Location Diagram

Instructions

Duration

approx. 30-60 minutes

Input

- **Software Distribution Diagram**
- **Baseline Solution Architecture Diagram**
- **Solution Context Diagram**
- **Solution Realization Diagram**

Why & What

Shows which of your architecture building blocks and solution building blocks are deployed at which locations. Highlights the interaction between geographically distributed building blocks.

With the help of the Environments & Location Diagram, you better understand the interaction between building blocks across different geographical locations. For this, you also add the locations of the users interacting with the aspired solution.

How to use it

1. Pick up the previously created Software Distribution Diagram and evolve it into the Environments & Location Diagram.

2. Identify the deployment environments in which your building blocks are running. Add more details to the “landing zones” of the Software Distribution Diagram by explicitly naming the data center providers, the data center location and infrastructure specific details per “landing zone”.

3. Map the solution building blocks, as defined in the Solution Realization Diagram, to the deployment environments.

4. Visualize the relationships of type request-response or information flow between the building blocks. This helps to understand network requirements .

Tips & Tricks

Consider the Solution Context Diagram, to pick up the existing IT systems and user’s interacting with the solution. For this, you can also look at the Baseline Solution Architecture Diagram.

The list of solution building blocks, which you will map to the locations and deployment environments, is taken from the Solution Realization Diagram.

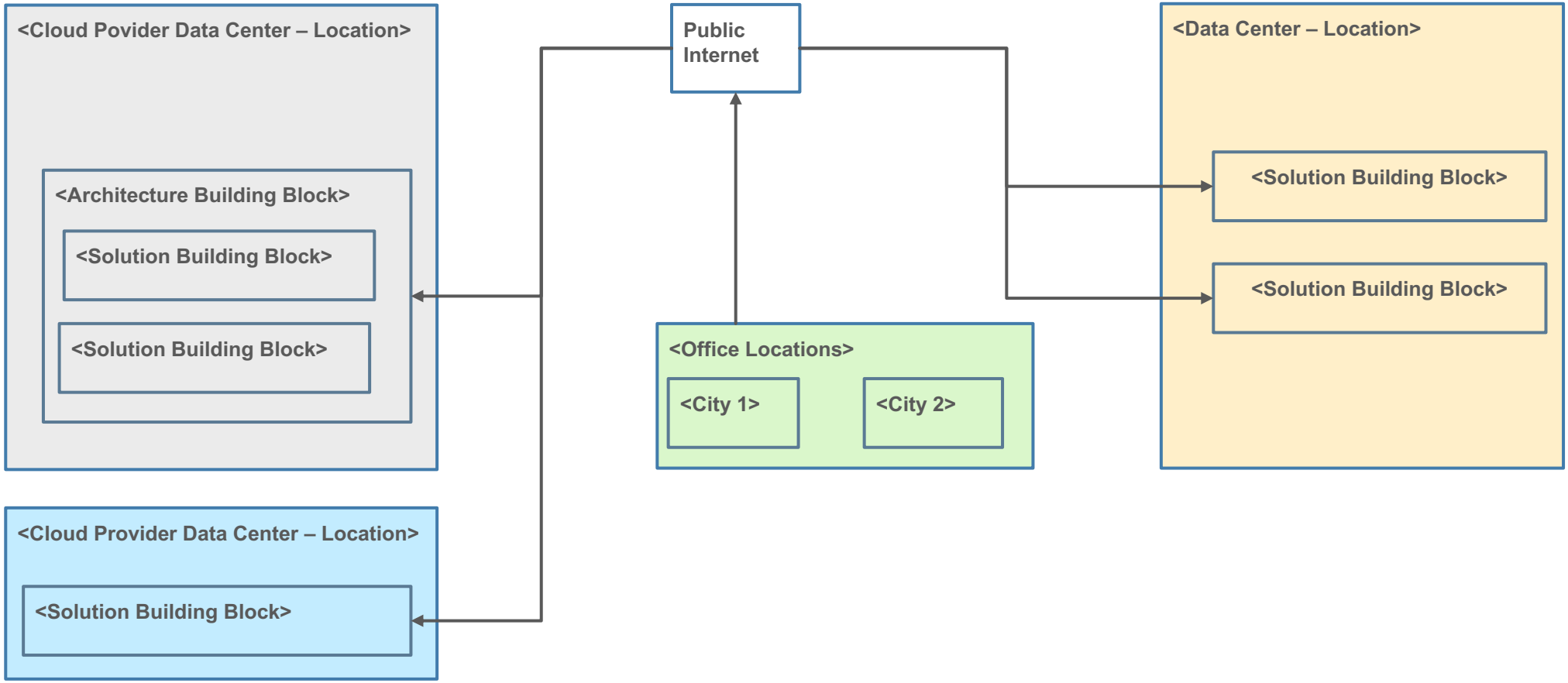
You can also outline different deployment environments such as development, test, and production.

Add technical details about the specific runtime environments of the solution building blocks, as well as network related details, in case they are relevant.

Remember to update your Architectural decisions.

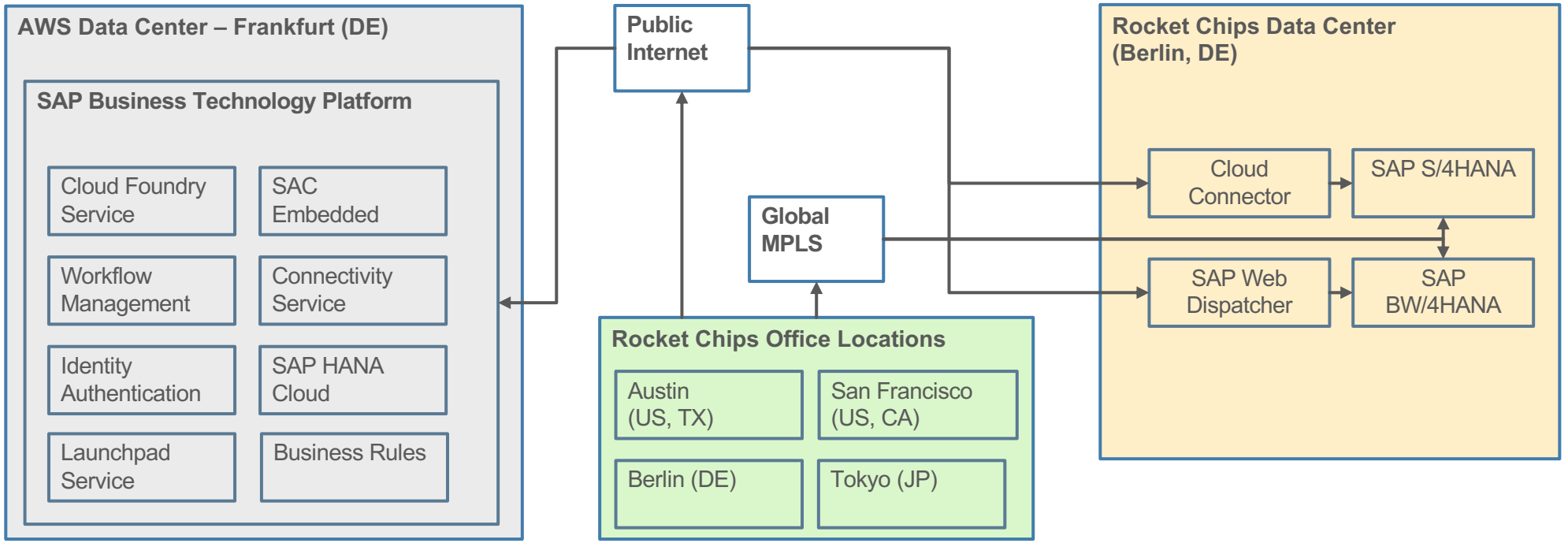
Environments & Location Diagram

Template



Environments & Location Diagram

Example



Legend:

Request response: ⟶

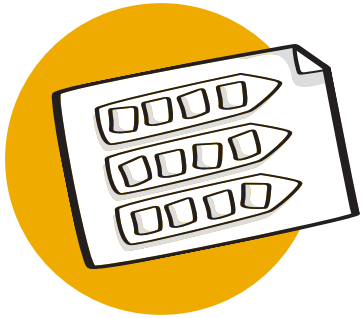
Information flow from source to target: ⟵ - - - -

Architecture

Roadmap

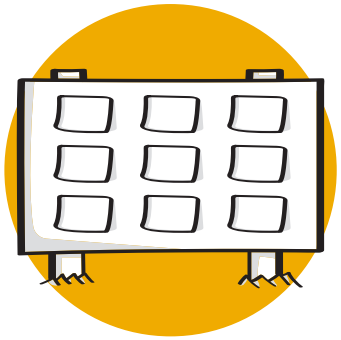
Instructions | Template | Example

Outlines actions that are required to realize the architecture



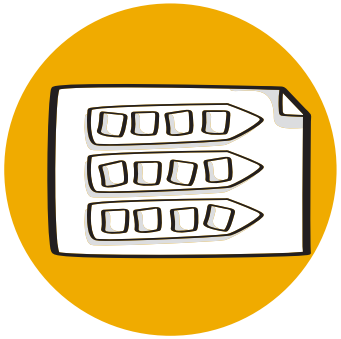
Architecture Roadmap

Actions that are required to realize the architecture.



Identify Work Packages

Identify individual work packages that will realize the aspired solution.



Define Sequence

Lay out the previously identified work packages on a timeline defining the sequence of implementation.

Highlight value

Highlight the business value of the individual work packages at each stage. Outline different sequential options for the implementation.

Architecture Roadmap

Instructions

Duration

approx. 60-120 minutes

Input

- **Solution Realization Diagram**
- **Solution Context Diagram**
- **Statement of Architecture Work**

Why & What

The Architecture Roadmap is an overview of specific actions that are required to realize the aspired solution. It lays out the individual actions on a timeline to show the sequence and progression from the baseline architecture to the target architecture. The purpose of the architecture roadmap is to discuss the optimal implementation sequence.

How to use it

1. Identify work packages required to realize the architecture. What needs to be done to implement the building blocks?

2. Categorize the work packages into different capabilities of the aspired solution - as defined in the Solution Context. This helps to correlate work packages to business value. Ideally, work packages within a capability category can be implemented mostly independently at their own speed.

3. Identify dependencies between work packages as they influence the sequence of implementation steps. Define the sequence and timeline of the implementation of the work packages.

4. Validate Architecture Roadmap by defining clear milestones and associated business value. Also, outline different sequential options for implementing the work products in case there are any.

Tips & Tricks

Input for creating the Architecture Roadmap is a mix of all the insights you have learned through the creation of the previous work products.

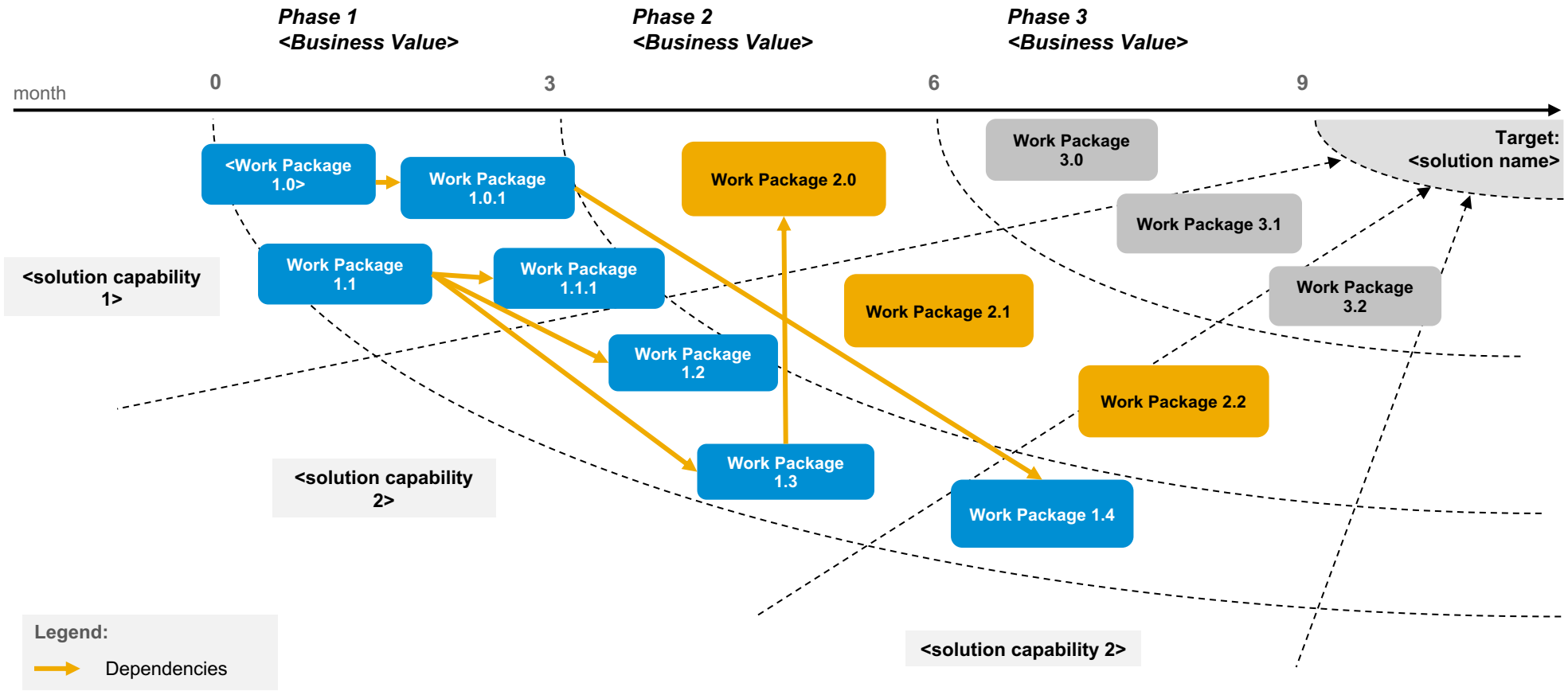
A good start to derive work packages is the Solution Realization Diagram. What needs to be done to implement the building blocks?

Think about pre-requisites that need to be in place before the implementation of work package can start. Is there any data access required? Is there a specific system integration required, for example?

Check the Statement of Architecture Work: Is the roadmap you have defined in sync with the scope of the Statement of Architecture Work?

You might want to create the Architecture Roadmap together with the team or partner implementing your architecture.

Architecture Roadmap
Template



Architecture Roadmap
Example