

IT architecture visualization

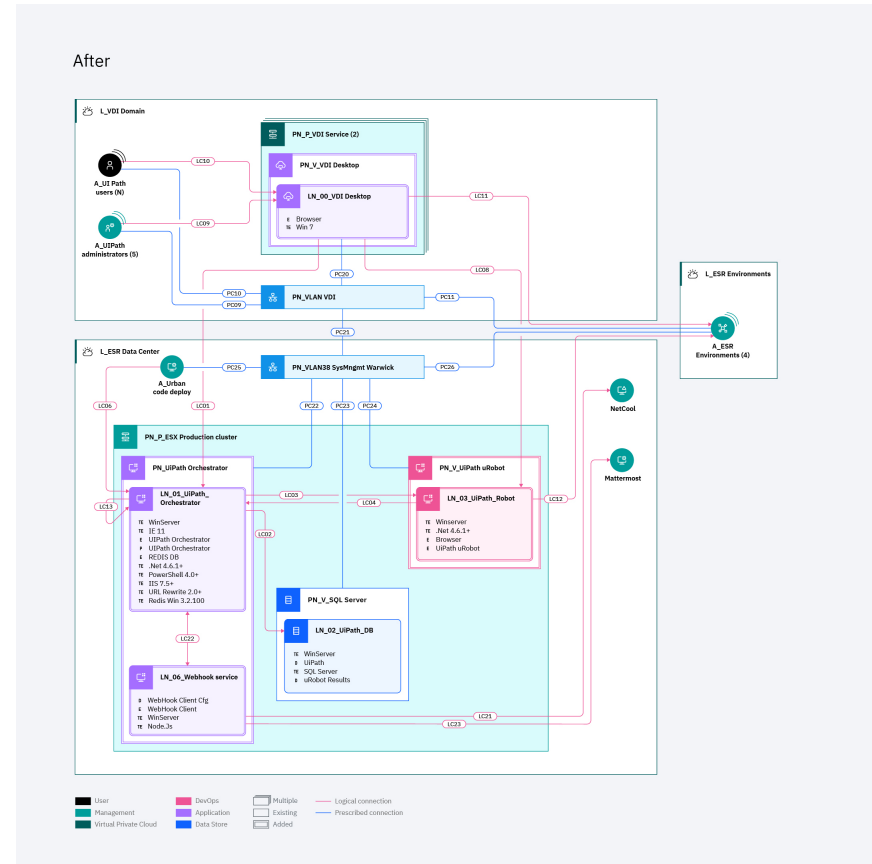
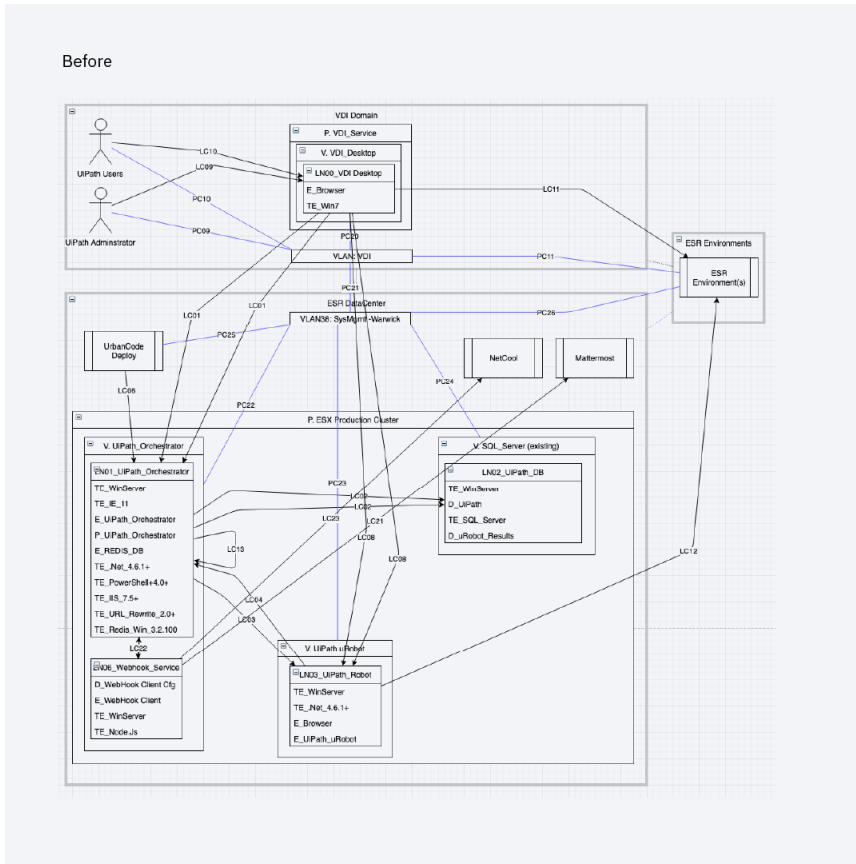
SDS/UMF meets IBM Design Language

- ↳ Overview
- ↳ Description of visual techniques
- ↳ Architectural views
- ↳ Other related views
- ↳ Resources

Overview

IBM's Unified Method Framework (UMF) is the modeling standard used by IBM for IT Architecture, and within that, the Systems Description Standard (SDS) covers the extensions required to describe the Operational Viewpoints.

The original visual aspects used for UMF and SDS, whilst they provided an accurate description of the function or operation of a system, are not visually appealing. The new visualization approach, described below, addresses this. It also allows for an additional depth of information to be conveyed within a diagram which outwardly appears to be of a similar or less complexity.

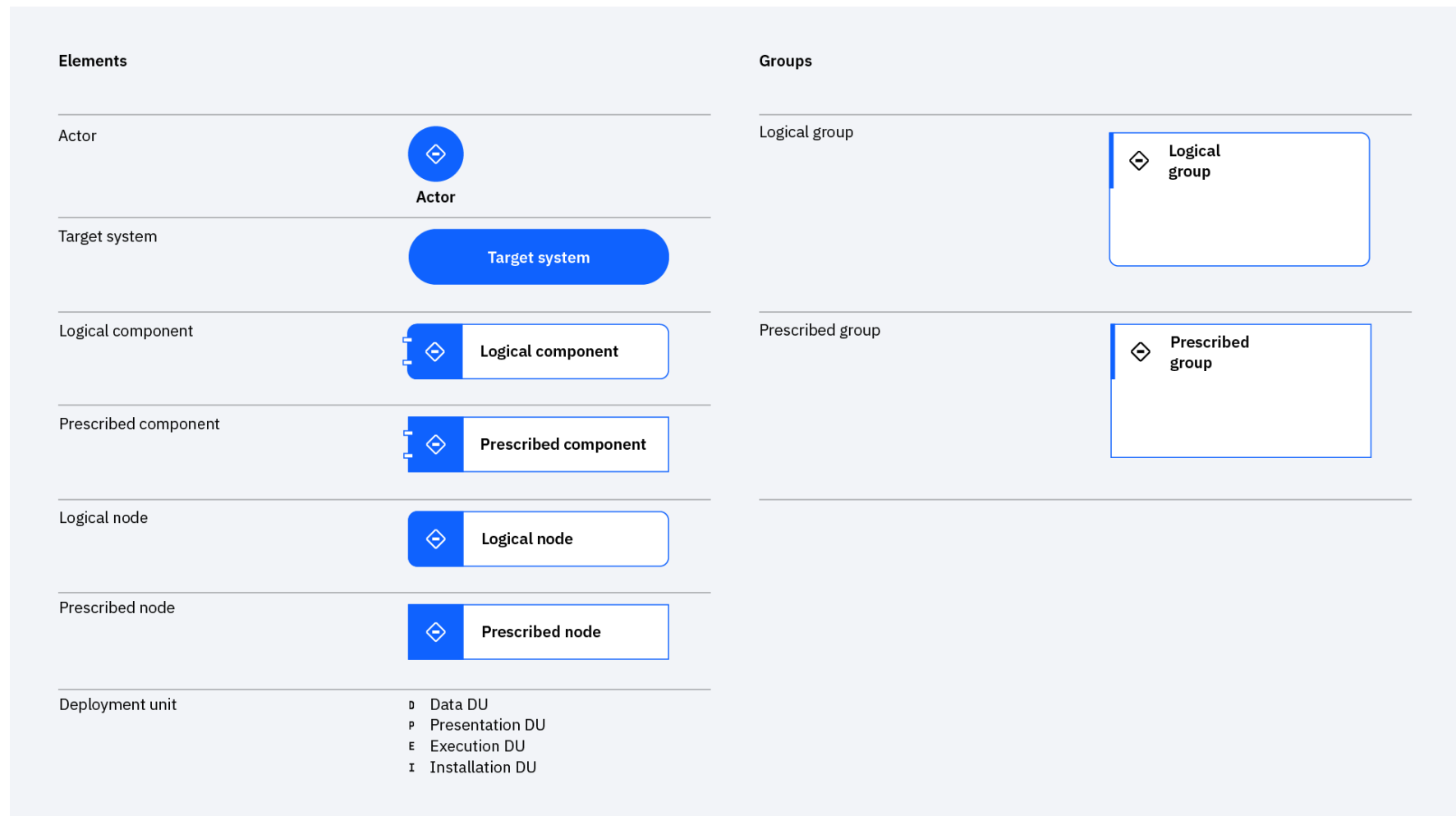


Description of visual techniques



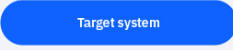

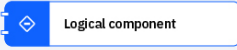
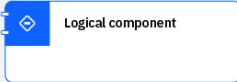




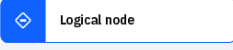
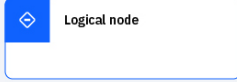

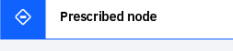
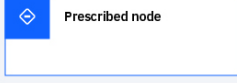
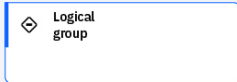
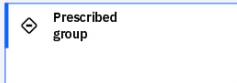
A number of visualization techniques are now used to convey meaning—these are outlined below.

Shape

The shape identifies the type of the element. Most of these elements can be used in three states: collapsed, expanded and expanded with nested elements.

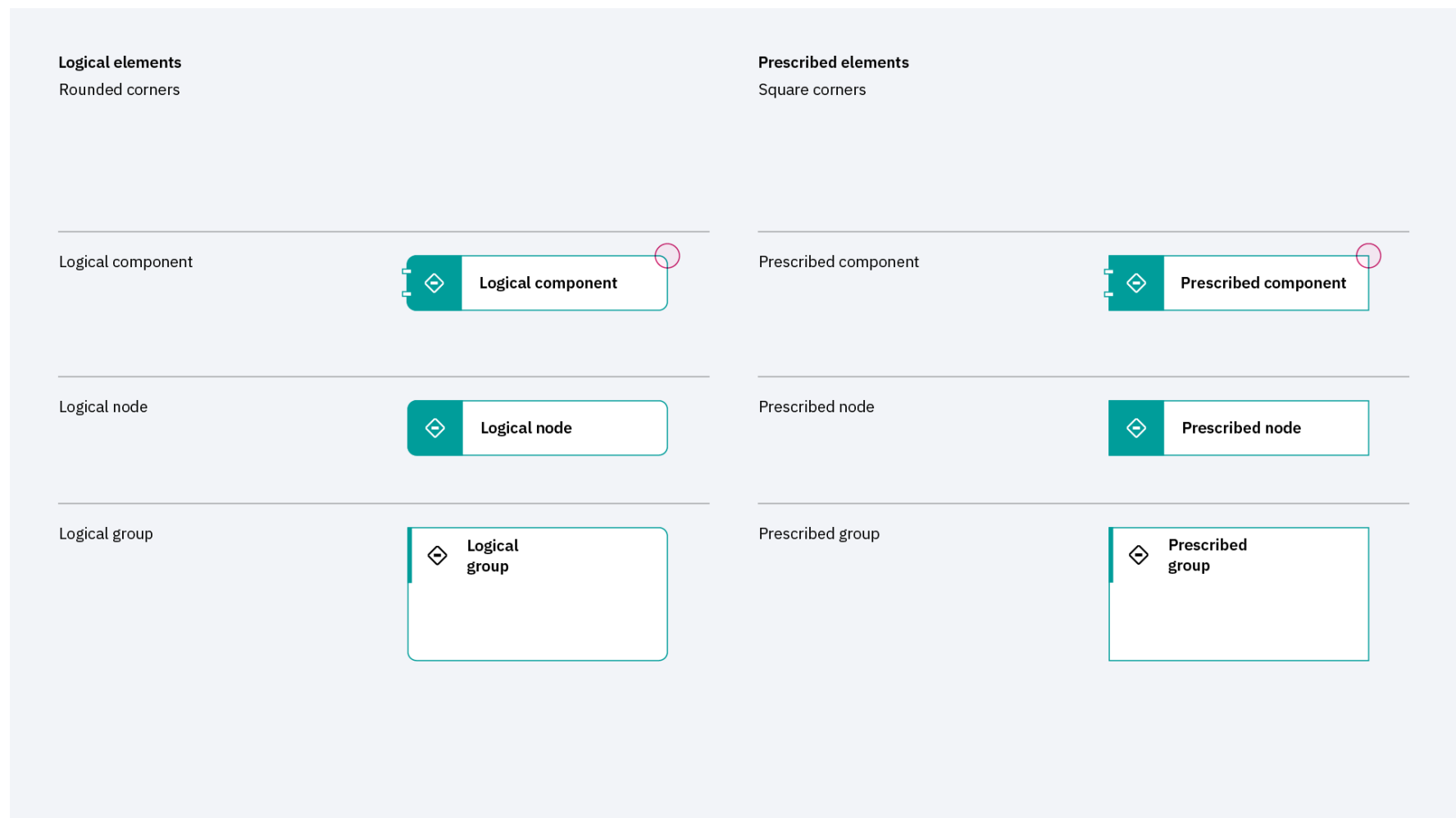


Most of these elements can be used in three states, as shown in the image below: collapsed, expanded and expanded with nested elements.

Elements	Collapsed	Expanded	Nesting
Actor	 Actor	N/A	N/A
Target system	 Target system		N/A
Logical component	 Logical component		
Prescribed component	 Prescribed component		
Logical node	 Logical node		
Prescribed node	 Prescribed node		
Logical group	N/A	N/A	
Prescribed group	N/A	N/A	

Corner

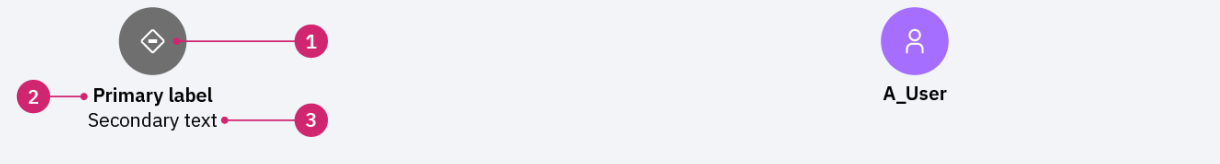
The components, nodes and groups can have rounded or squared corners. This identifies the level of abstraction of the element, rounded being Logical, square being Prescribed.



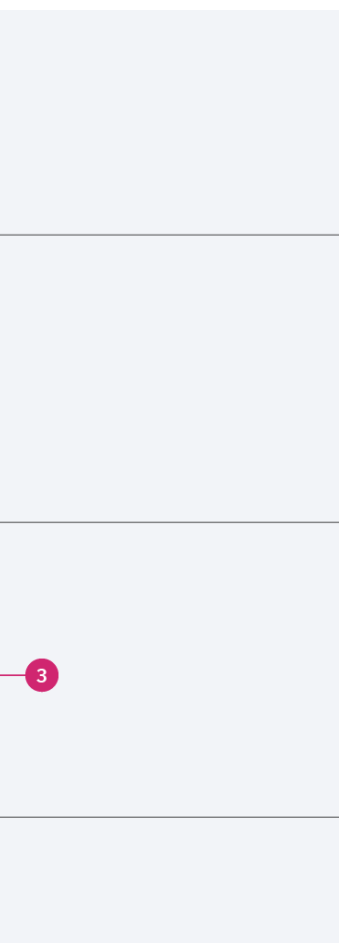
Textual identifier

Also identifies the type of element by prefix e.g. A_, N_, PN_, etc., along with the specific instance of the element. This is unchanged from the previous/legacy visualization technique.

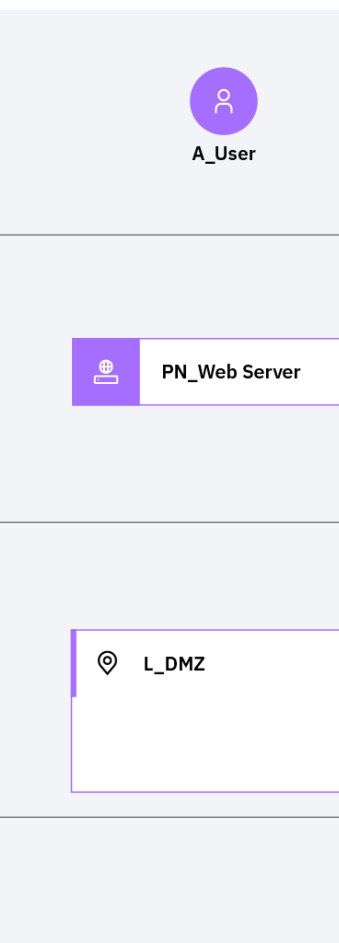
Collapsed elements



Expanded elements



Groups















1. Icon, optional for expanded elements and groups
2. Primary label: Identifies the type of element
3. Secondary text: Can be used to show the code of the element or to include secondary information about the element.

Connectors

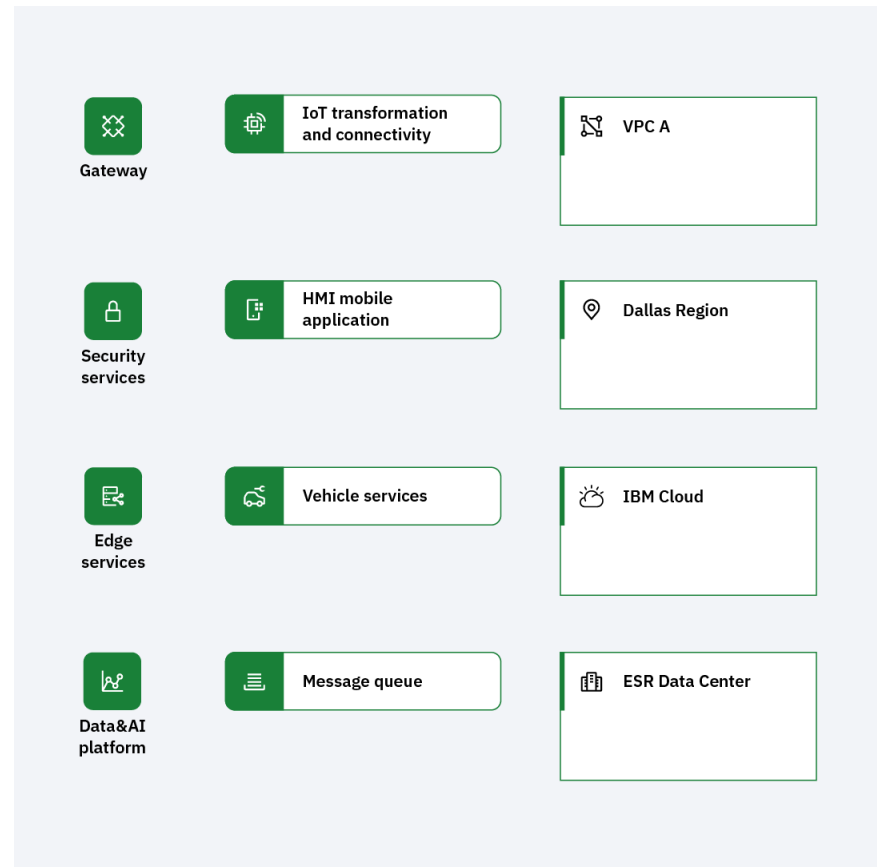
Connectors represent relationships between elements. Depending on the type of diagram, a number of line styles and endings can be used to represent various types of relationships.

Connectors can also use different colors. However, to avoid color blindness accessibility issues, it's best to use color in combination with labels or line styles so that any difference can be accessible to all users. Since line styles and color can have different meanings, depending on the diagram, be sure to include the definitions in the legend.

Solid		T-shape	
Dashed 1		Arrow, fill	
Dashed 2		Circle, fill	
Dotted		Circle, open	
Double		Diamond, fill	
Tunnel		Diamond, open	

Icon

Using an icon is optional. An icon identifies the type of element which is shown, for example a Firewall, or a Cloud Zone. A practitioner should use an existing **IBM Design Language** icon where possible, although additional icons can be used where necessary.



Color

Using multiple colors in a diagram is optional—however it is a good way to make them easier to interpret. Color can be applied to each element. The color palette shown below is the recommended palette for IT architecture diagrams. It is a subset of the IBM Design Language [color palette](#).

Primary colors are accessible on light backgrounds. Use primary colors for element outlines, side bars, color blocks, badges and connectors. Use secondary colors only for fills. Keep all elements monochrome (e.g. a red primary color can only be used with a red or white fill)

Primary colors

Red 50 fa4d56	Red 70 a2191f
Magenta 50 ee5396	Magenta 70 9f1853
Purple 50 a56eff	Purple 70 6929c4
Blue 60 0f62fe	Blue 80 002d9c
Cyan 50 1192e8	Cyan 70 00539a
Teal 50 009d9a	Teal 70 005d5d
Green 60 198038	Green 80 044317
Cool Gray 50 878d96	Cool Gray 70 4d5358
	Black 000000

Limited use

Yellow 50 b28600
Orange 50 eb6200

Secondary colors

Red 10 fff1f1
Magenta 10 fff0f7
Purple 10 f6f2ff
Blue 10 edf5ff
Cyan 10 e5f6ff
Teal 10 d9fbfb
Green 10 defbe6
Cool Gray 10 f2f4f8
White ffffff

Limited use

Yellow 10 fcf4d6
Orange 10 fff2e8

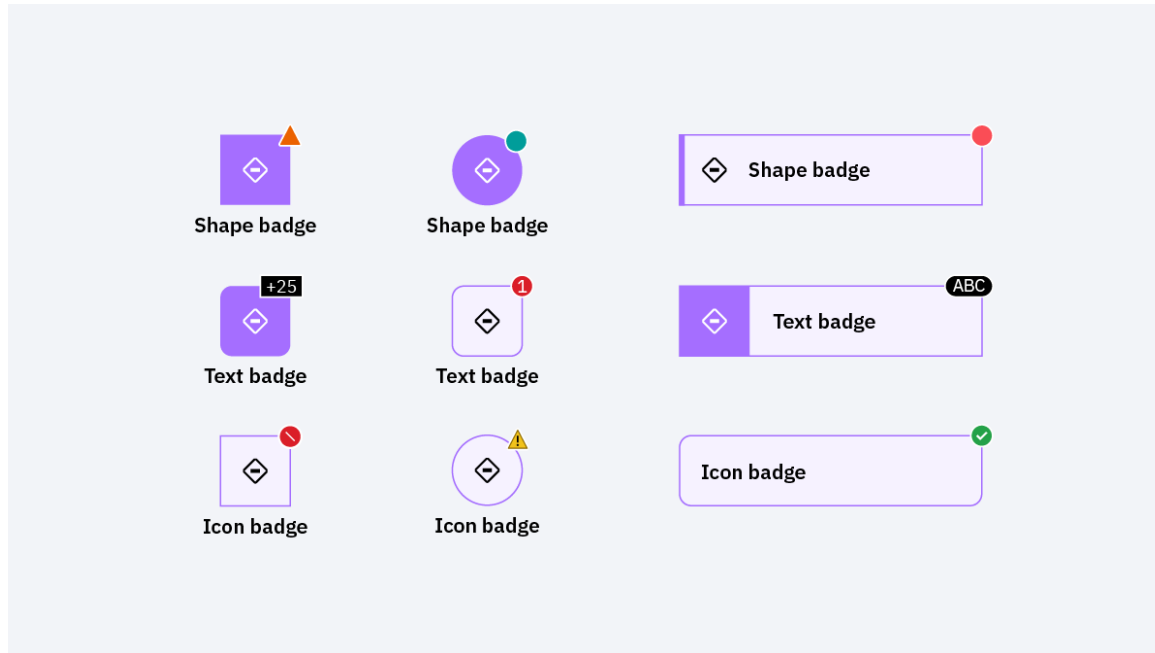
Whilst color can be used to convey any meaning that the Architect wishes, normally it is recommended that it is used to represent the *domain* of an element—e.g. Red for a Security component. This is illustrated below as the *Recommended color key*. Always include the colors used in the legend, even when the recommended color key is used.

Recommended color key

Actor Black
Security Red 50
DevOps Magenta 50
Application Purple 50
Data store Blue 60
Network Cyan 50
Management Teal 50
Virtual Private Cloud Teal 70
Compute Green 60
Other Cool Gray 50

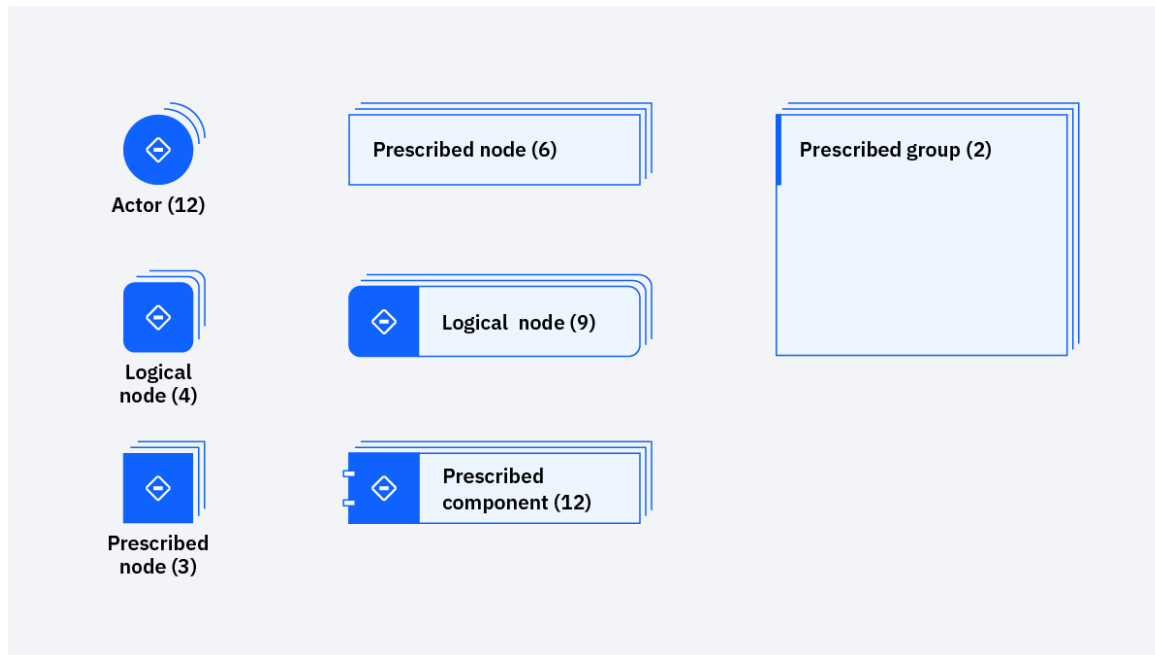
Badge

A badge (or tag) is a small shape that can be added at the top right of an element to convey further information. It is recommended that this is used to convey different information to that portrayed using other techniques described above, for example it is not good practice to use a badge to identify that a node is Prescribed rather than Logical. Using a badge is optional.



Multiple instances

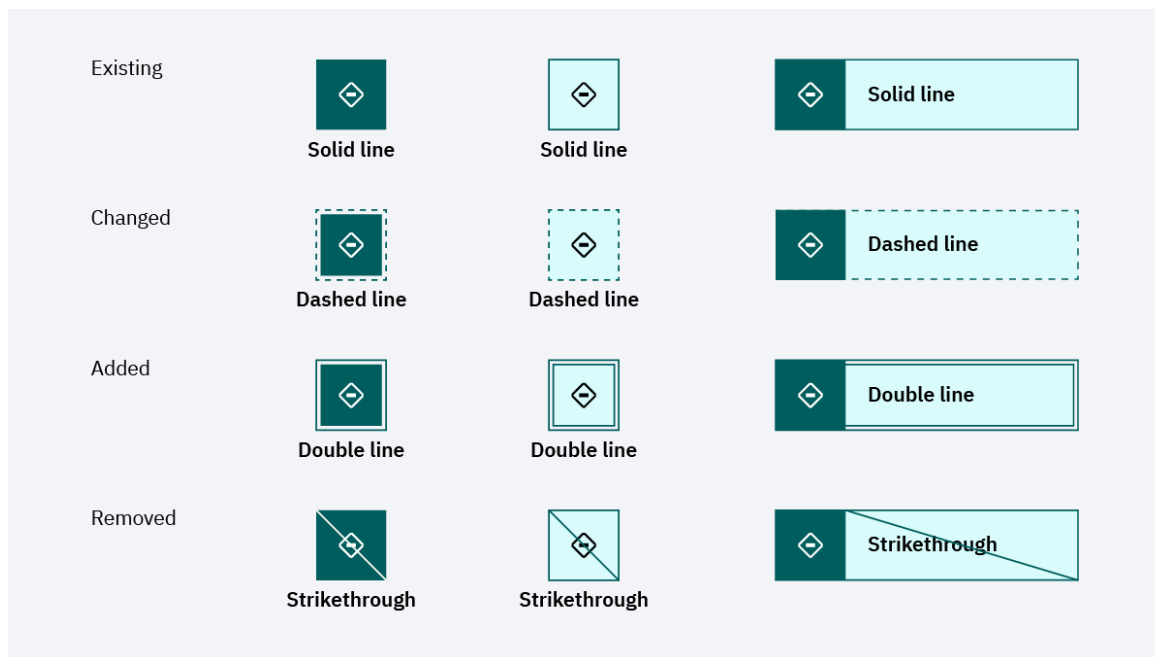
To represent multiple instances of the same element, an offset effect can be used, as shown here. Always show an additional two offset lines, regardless of the actual number of repetitions. Mention the number of instances in parentheses, in the text of the element.



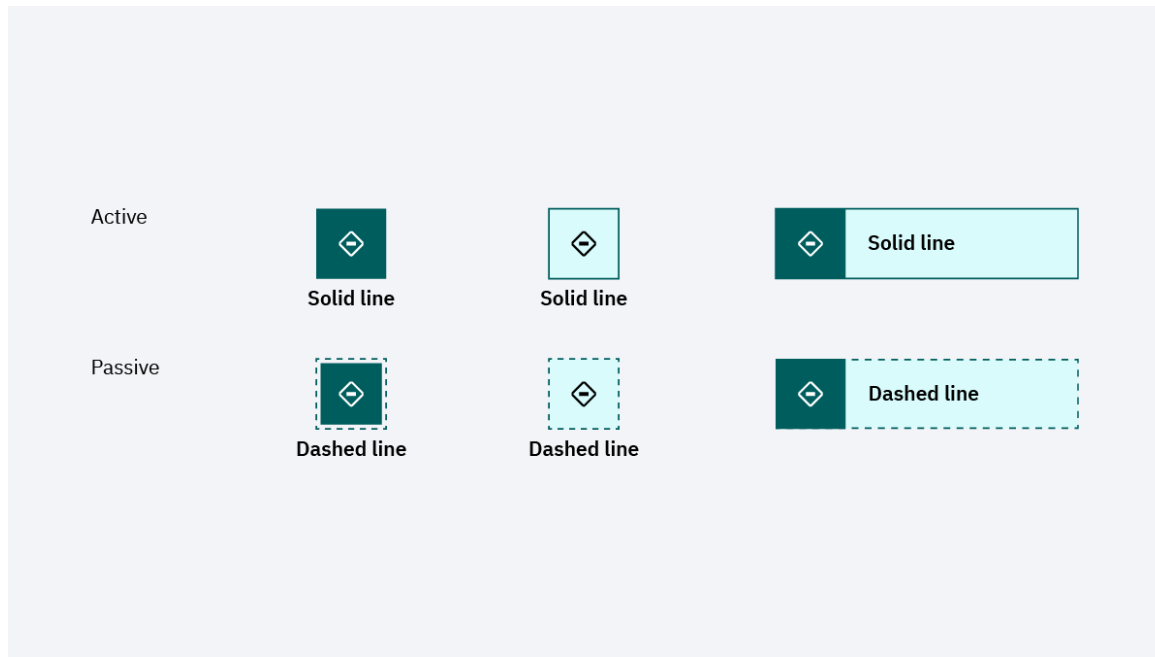
Style

Line styles can be applied to the elements to convey different types of meaning. The most common are:

1. Change: A technique is recommended which allows change to be identified on a diagram. Where it is used, for a new release of a system, this identifies which items have been affected by the release.



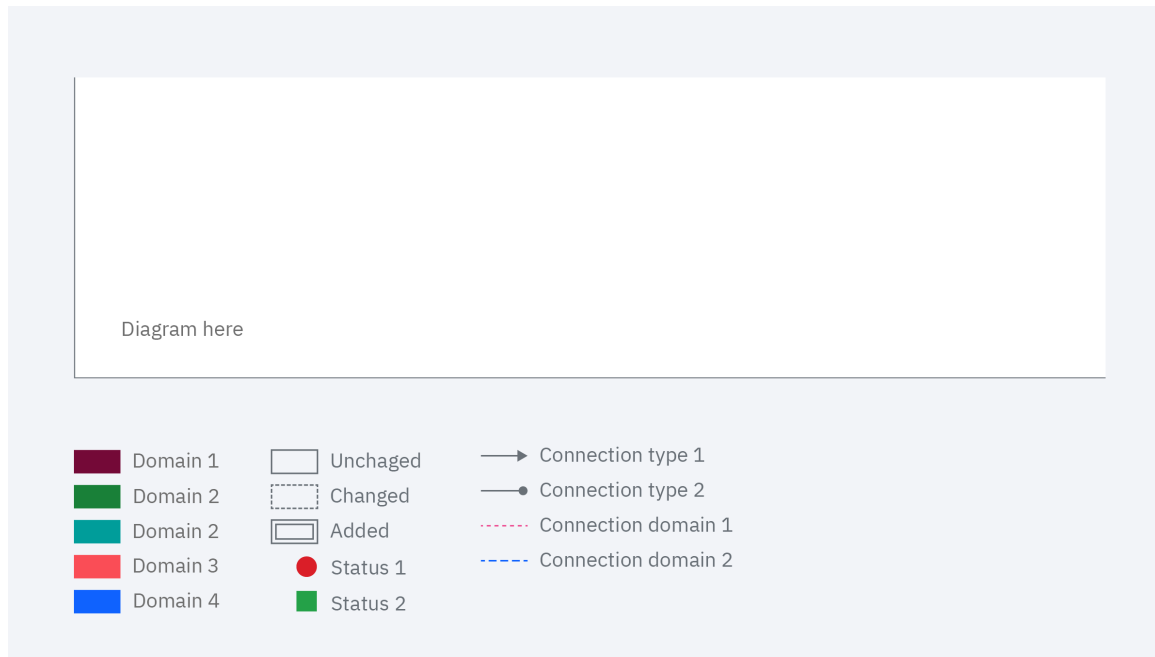
2. Active/Passive: Elements which may be in different states can also be depicted using line styles.



Always include a legend with the definition of the line style used, even when using one of the above recommendations.

Legend

Always include a legend to identify how you have made use of visual techniques, like Colour, Badges, Style.

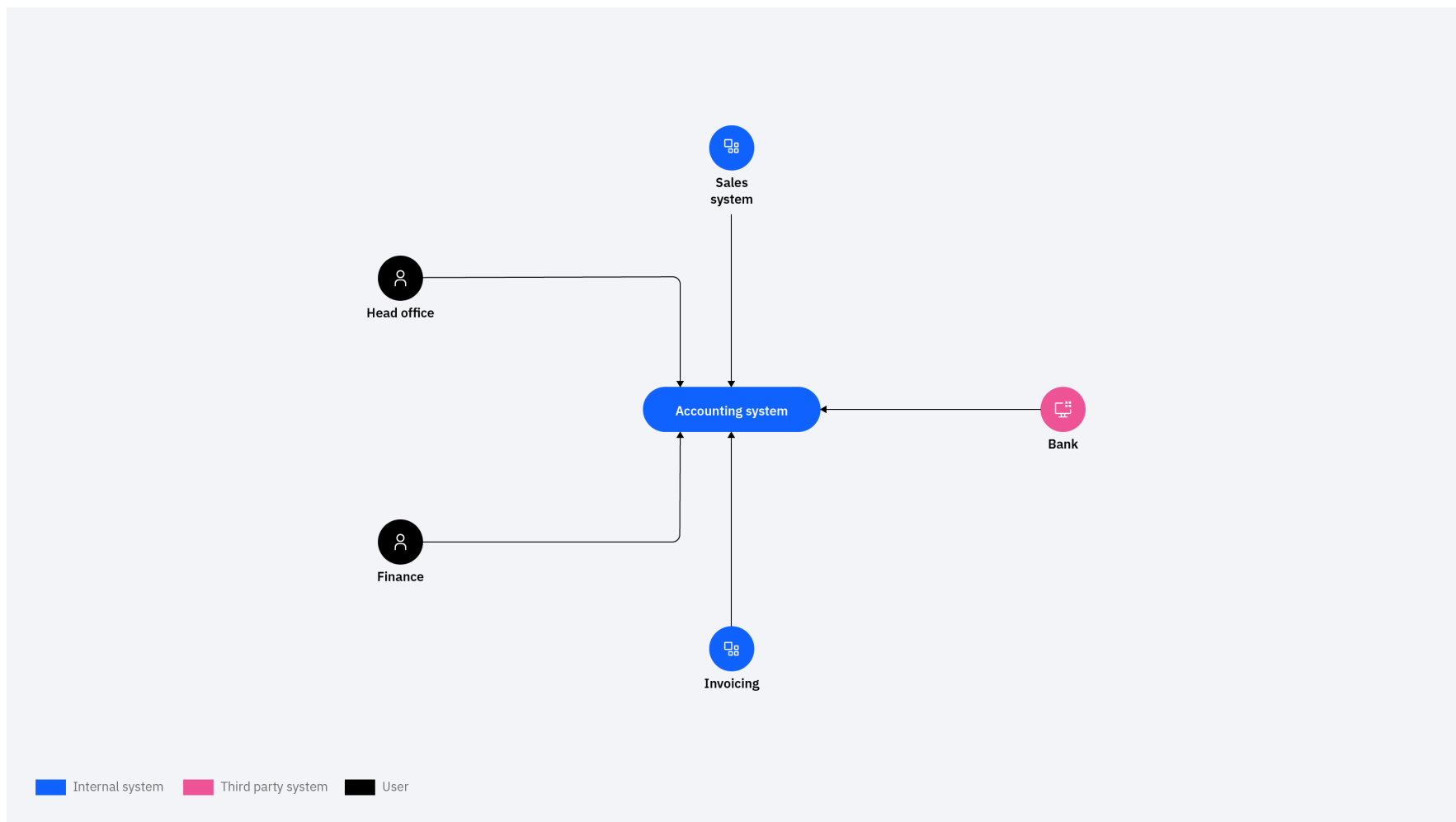


Architectural views

There are a number of architectural views which use the above techniques. These are described below.

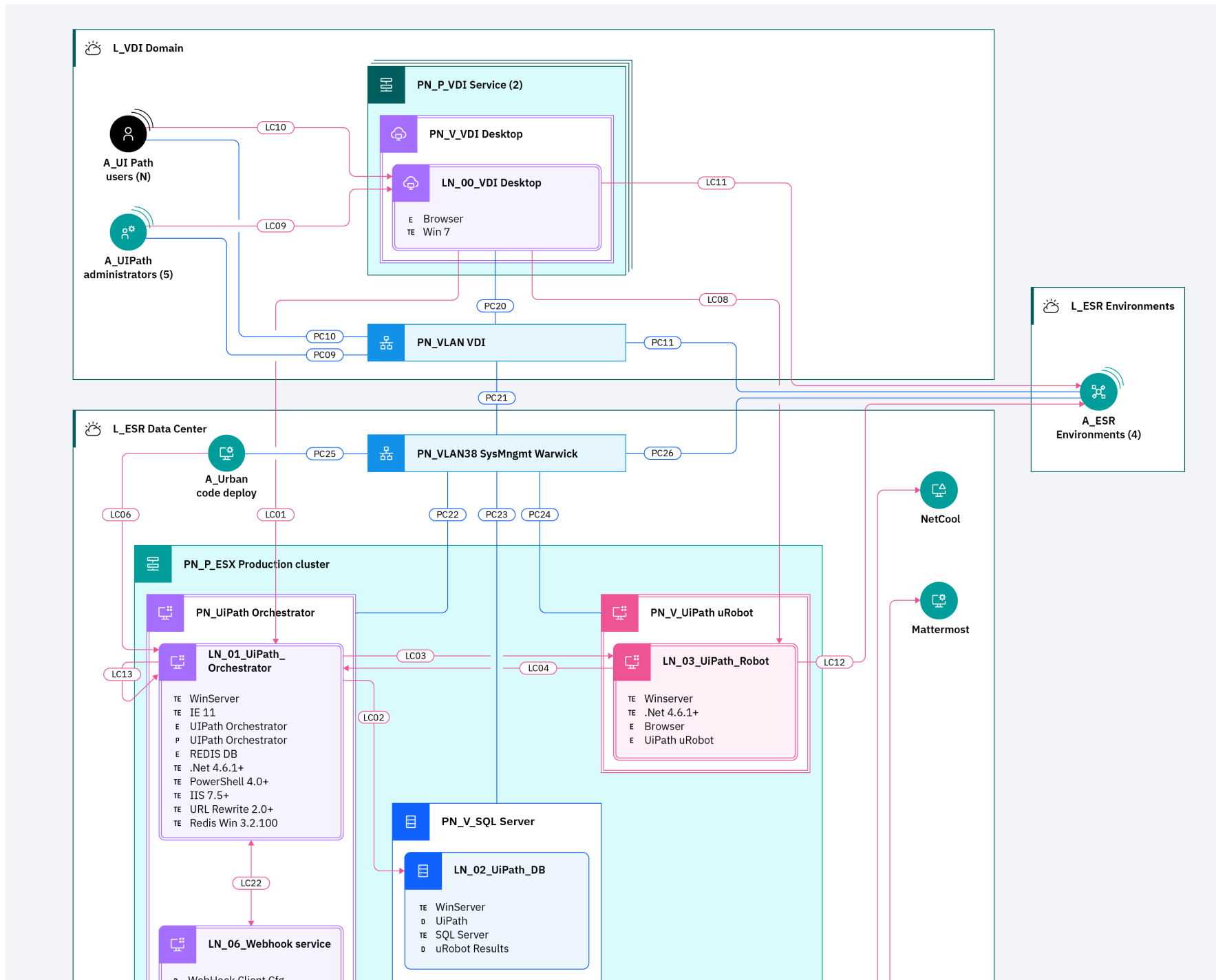
System context diagram

The systems context diagram describes the relationship between the target IT system and any actors. More detail on the systems context diagram can be found [here](#).

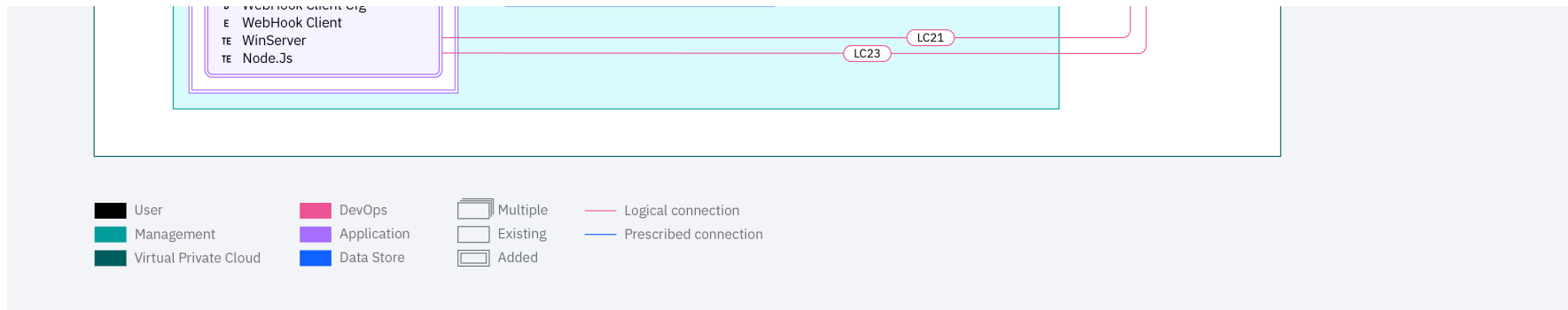


Operational model

The operational model describes the interaction of nodes within the target IT system and with external actors. It can be logical or prescribed. More detail on the operational model can be found [here](#).

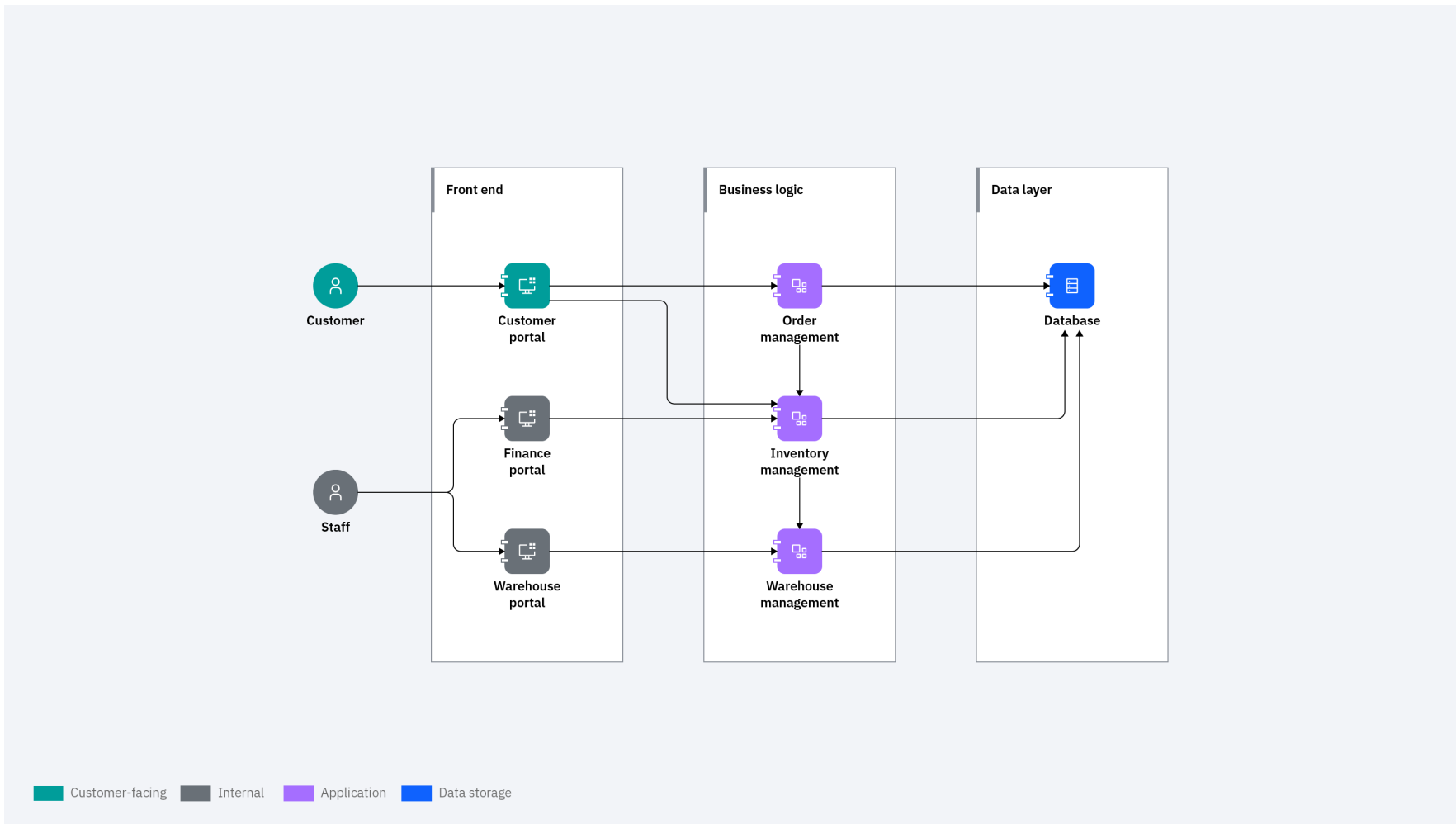


0 am



Component model

The component model describes the interaction of components within the target IT system and to external actors. More details on the component model can be found [here](#).

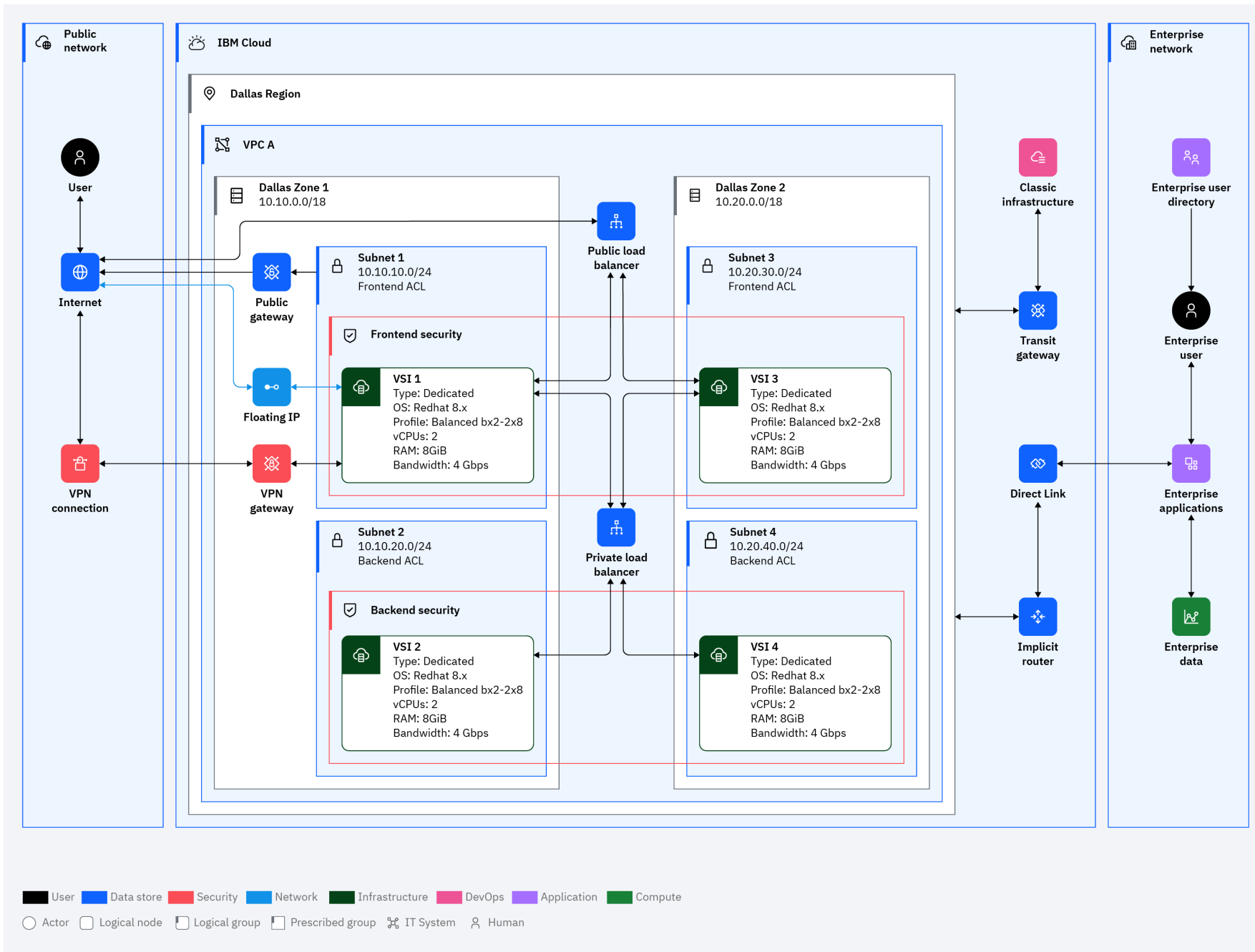


Other related views

Whilst not SDS, the same visual techniques are used in other IT architectural diagrams.

Cloud Architecture Center

The [Cloud Architecture Center](#) provides information on building apps on the cloud. The reference architectures define the basic pattern, while implementations provide specific technology, practices, and tool choices to build and deploy that pattern. The reference architectures are now using the same visualization techniques described above.



Resources

Tools and kits

[Cognitive Architect](#)

[Diagrams.net \(ex draw.io\)](#)

IBM Design Language

[Iconography](#)

[Color](#)

[Technical diagrams](#)

Methods and education

[IBM Architect Profession Education](#)

[GBS Method](#)

[Mosaic](#) (Method for complex systems integration)

[Team Solutions Design](#) (Method for client engagements)

Support

Slack: [#it-architecture-diagrams](#)

0 comments

 Privacy Settings

There are no comments.