IT architecture visualisation

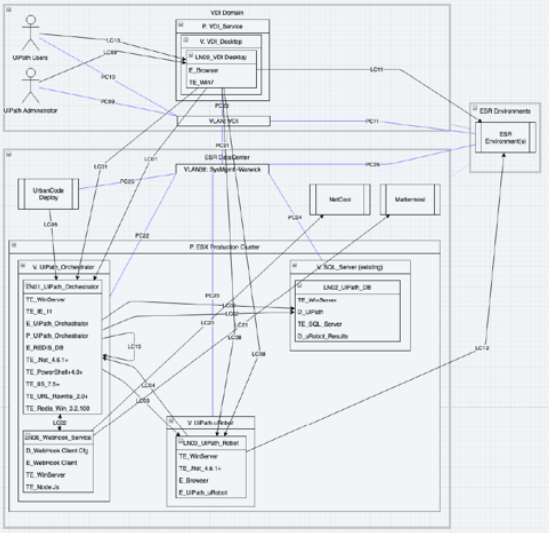
SDS/UMF meets IBM Design Language

Overview

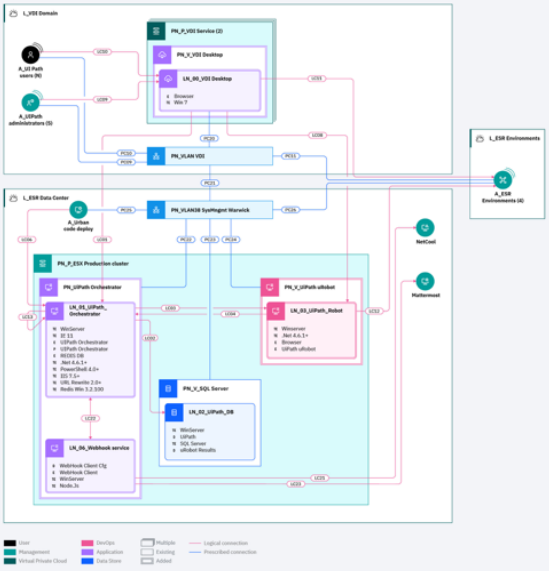
IBM's Unified Method Framework (UMF) is the modeling standard used by IBM for IT Architecture, and within that, the Systems Description Standard (SDS) covers the extensions required to describe the Operational Viewpoints.

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.

Before



After

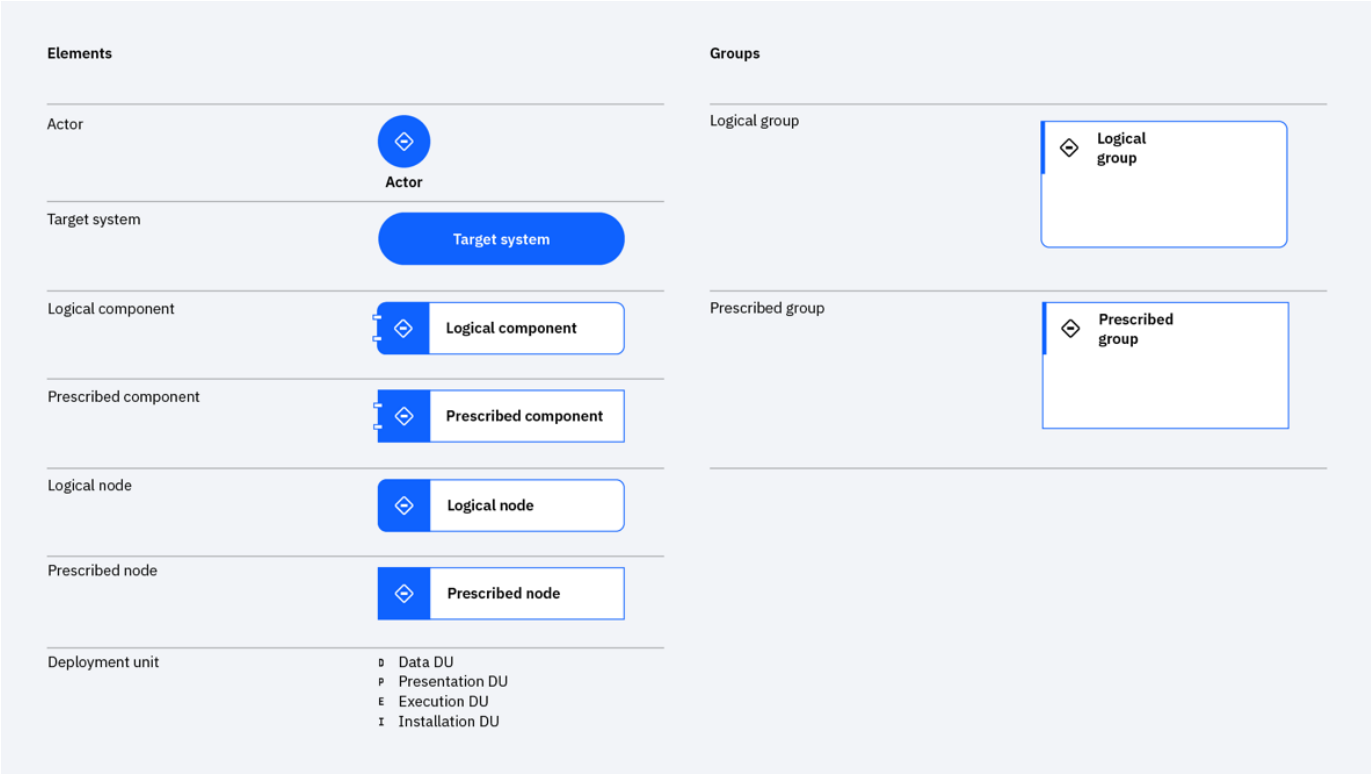


Description of visual techniques

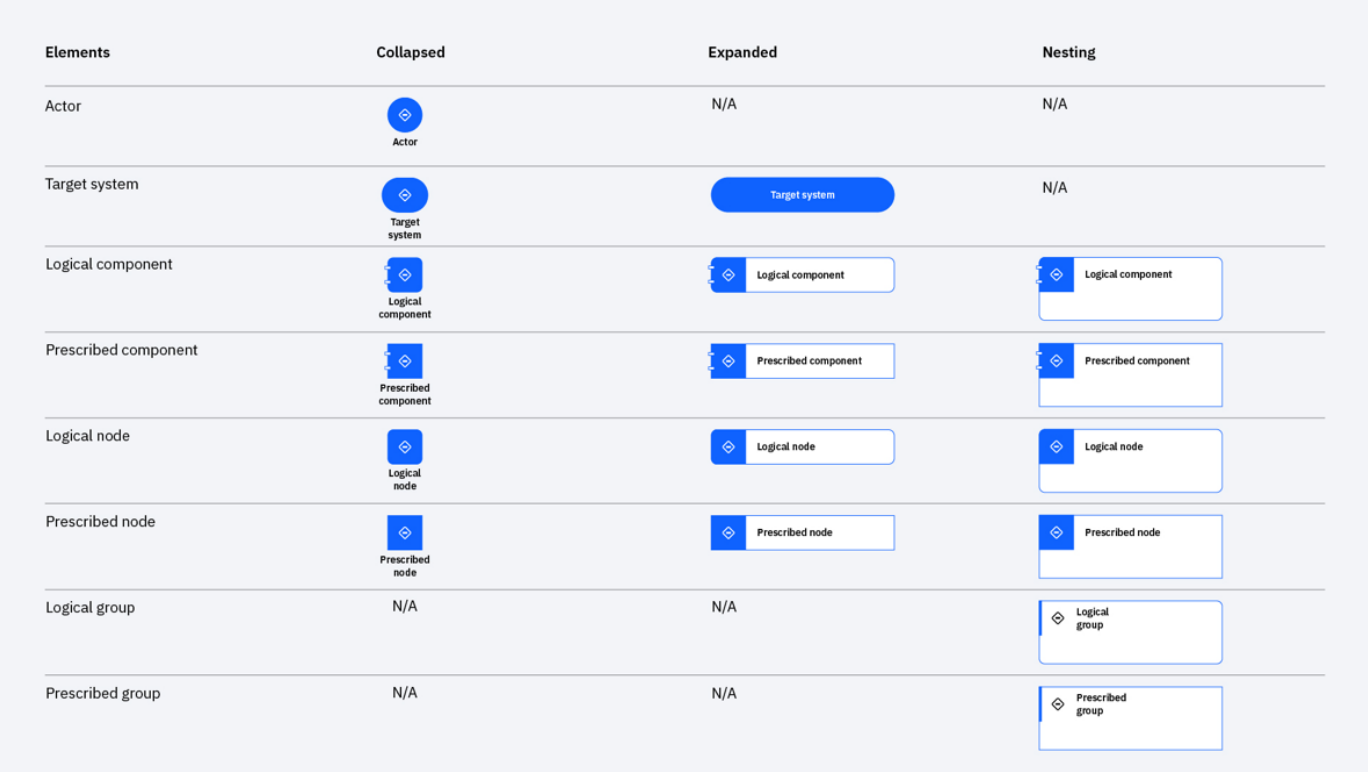
A number of visualization techniques are now used to convey meaning—these are outlined below.

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.

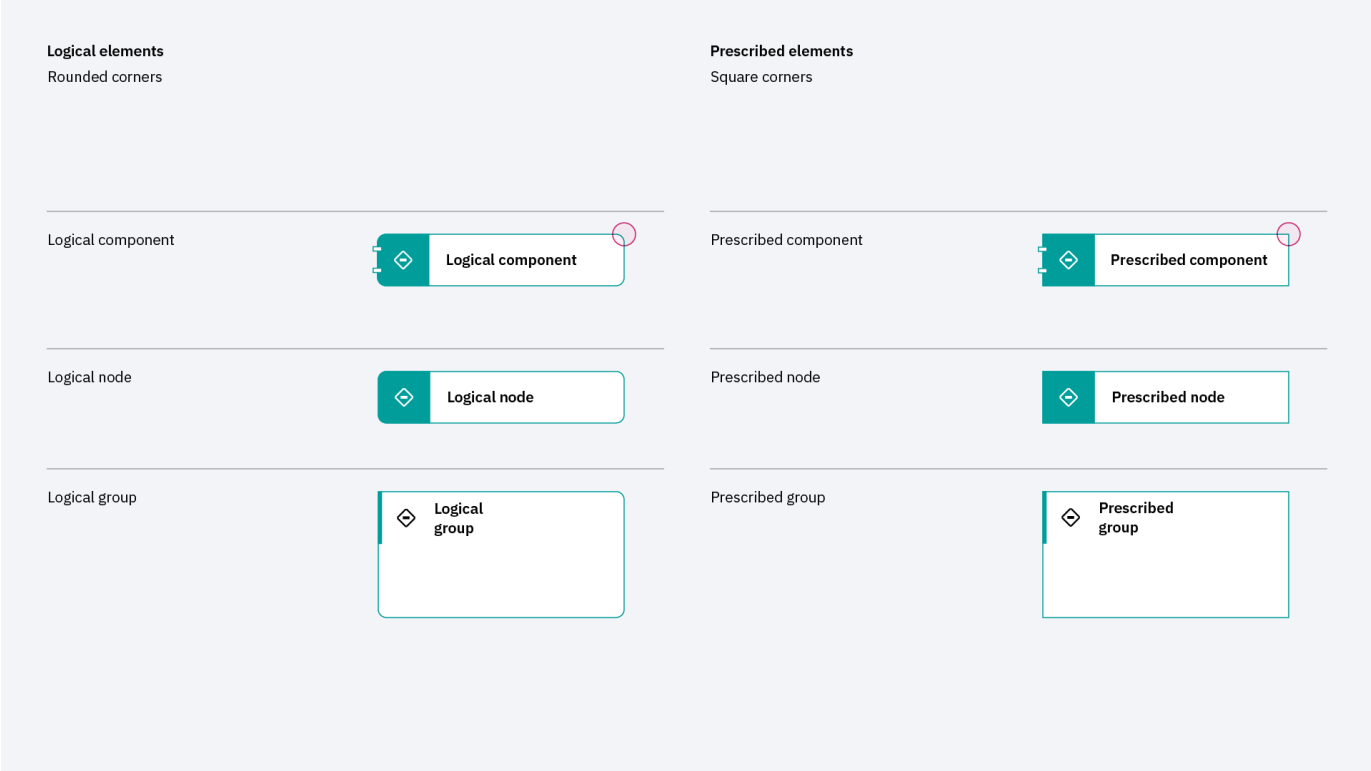


Most of these elements can be used in three states, as shown in the image below: collapsed, expanded and expanded with nested elements.



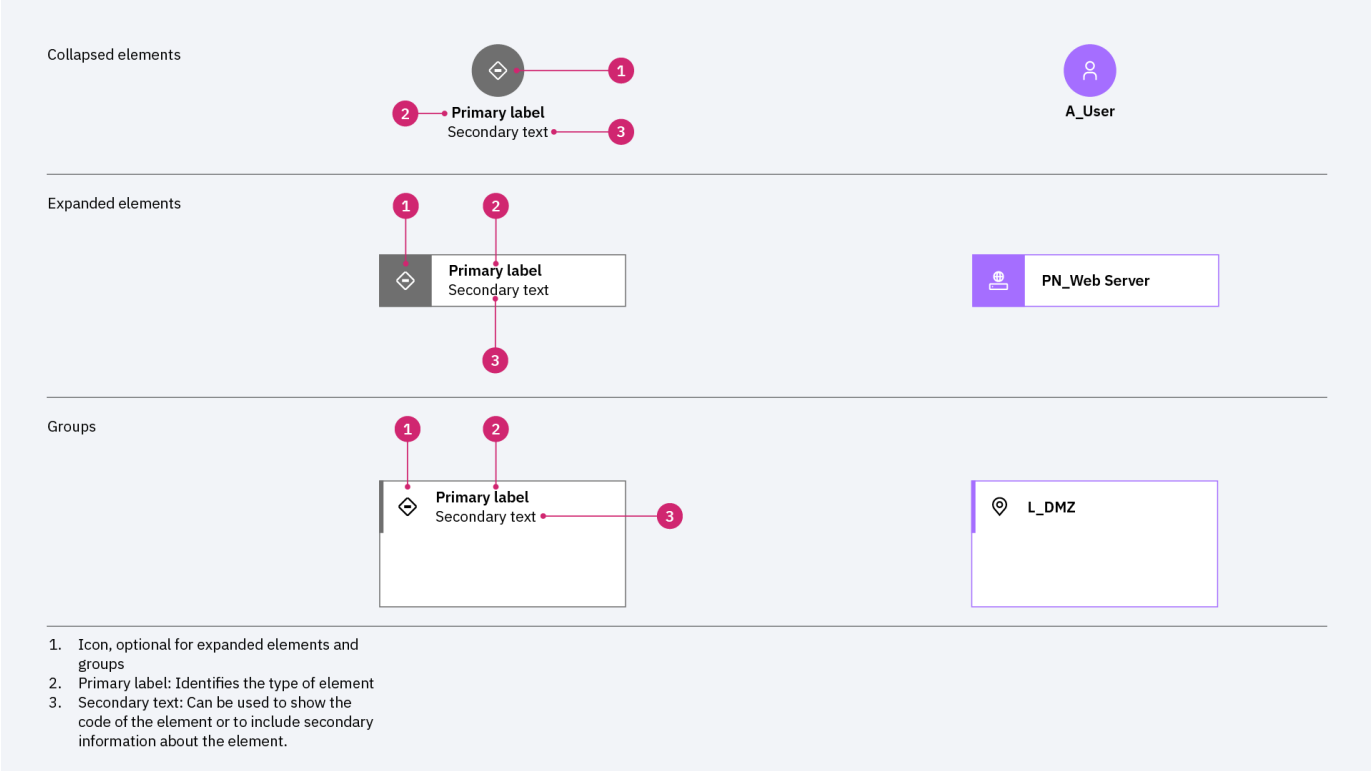
Corner

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.



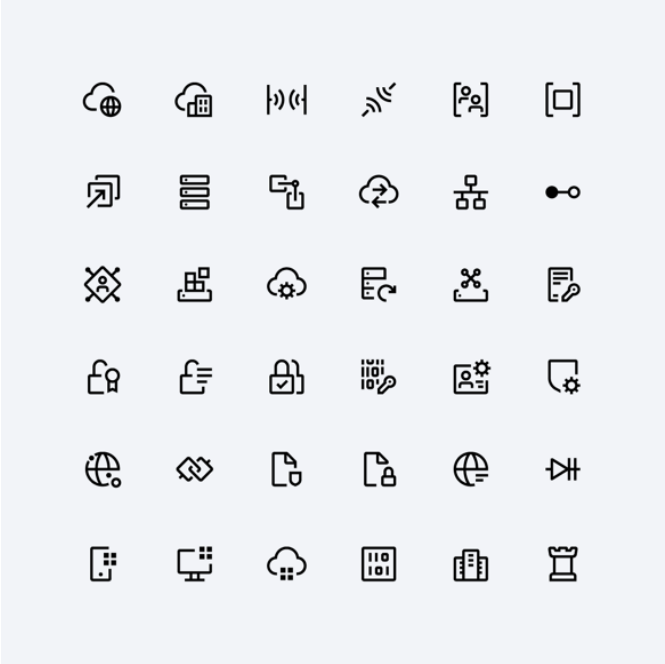
Textual identifier

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.



Icon

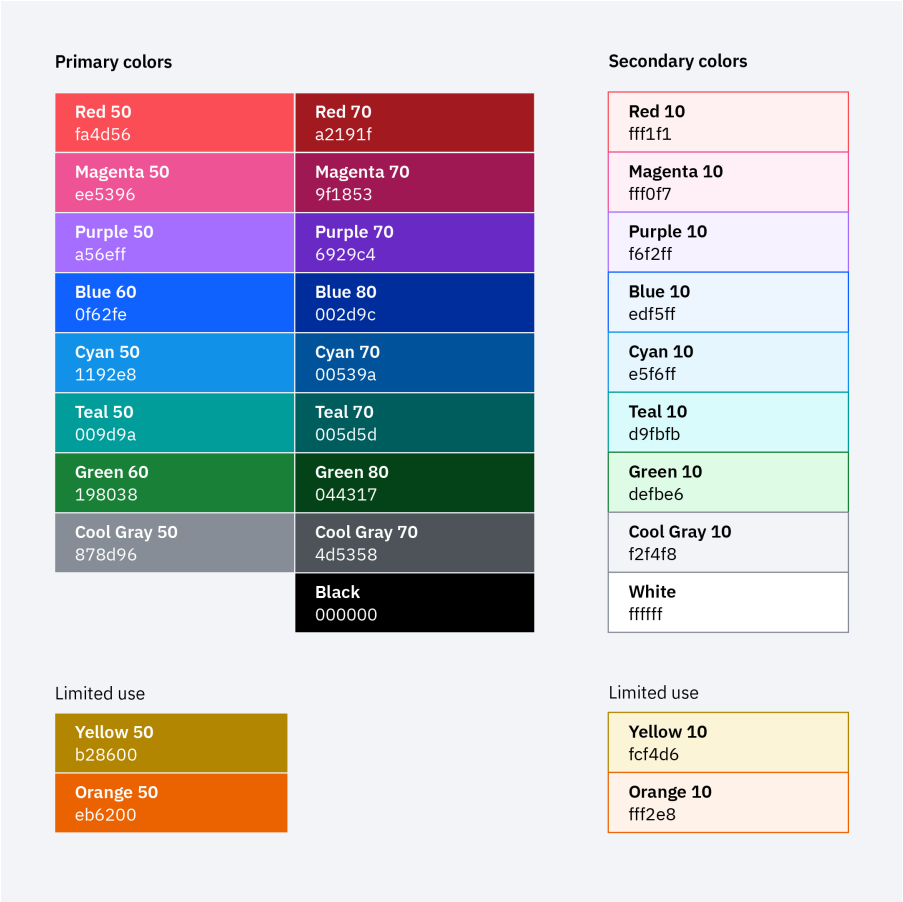
Using an icon is optional. An icon identifies the type of element which is shown, for example a Firewall, or a Cloud Zone. A practitioner should use an existing IBM Design Language icon where possible, although additional icons can be used where necessary.



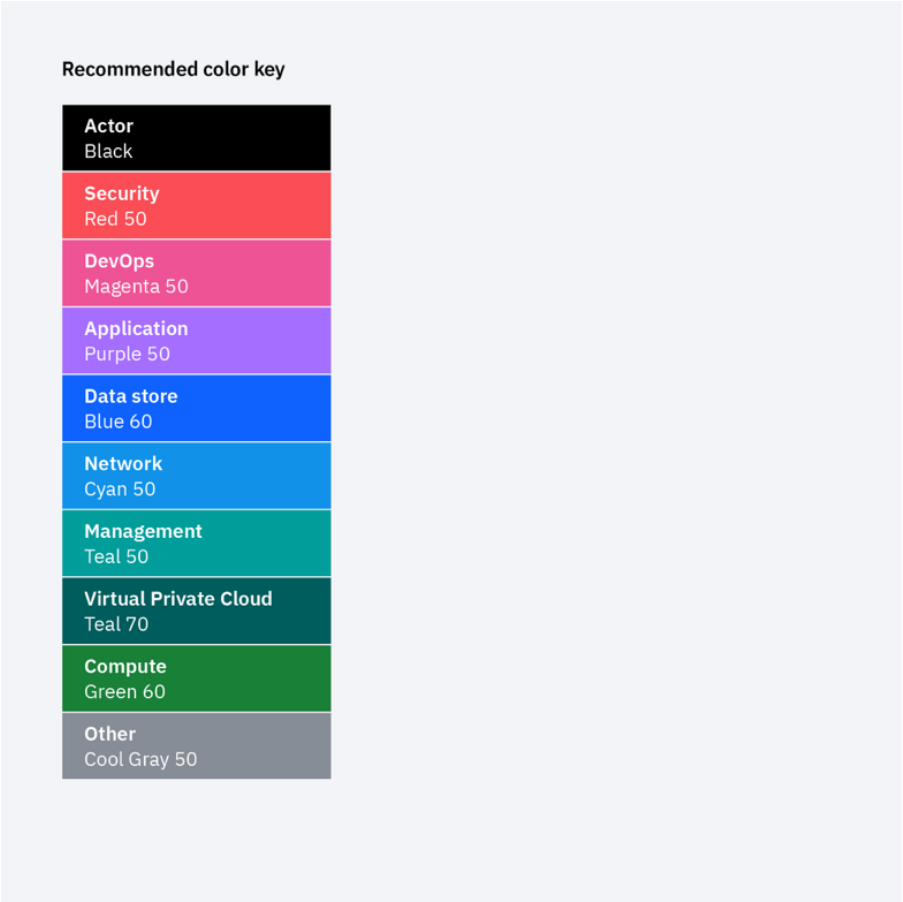
Color

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)

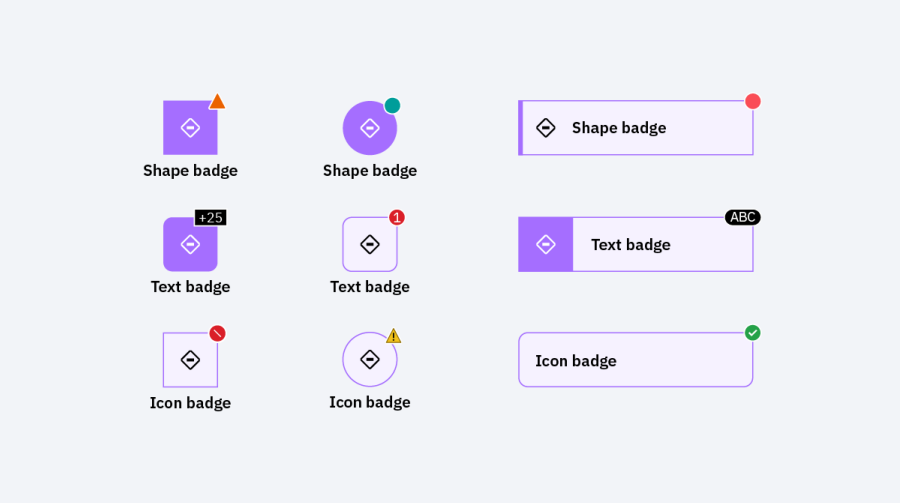


Whilst color can be used to convey any meaning that the Architect wishes, normally it is recommended that it is used to represent the *domain* of an element—e.g. Red for a Security component. This is illustrated below as the *Recommended color key*. Always include the colors used in the legend, even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



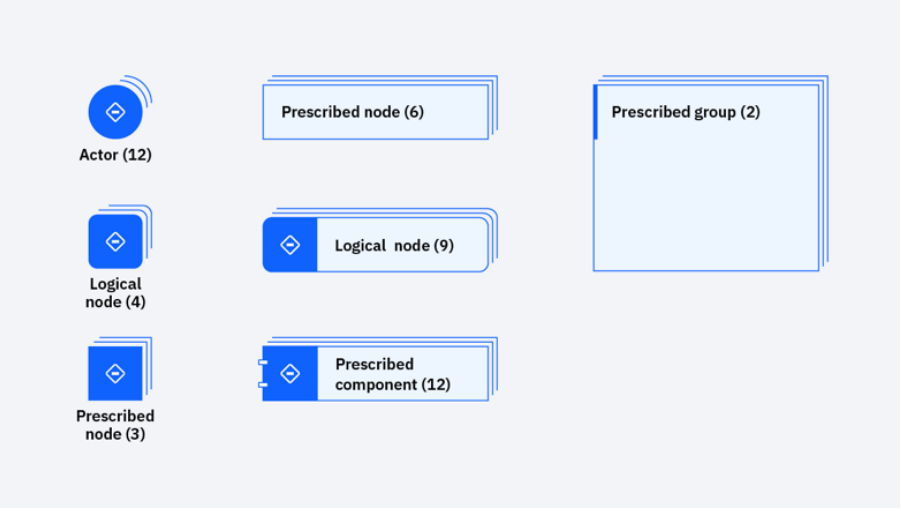
Badge

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

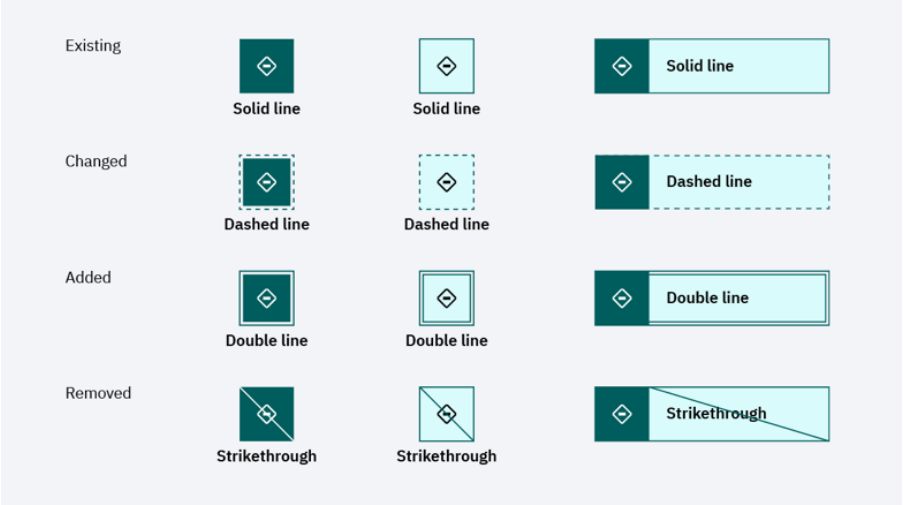
To represent multiple instances of the same element, an offset effect can be used, as shown here. Always show an additional two offset lines, regardless of the actual number of repetitions. Mention the number of instances in parentheses, in the text of the element.



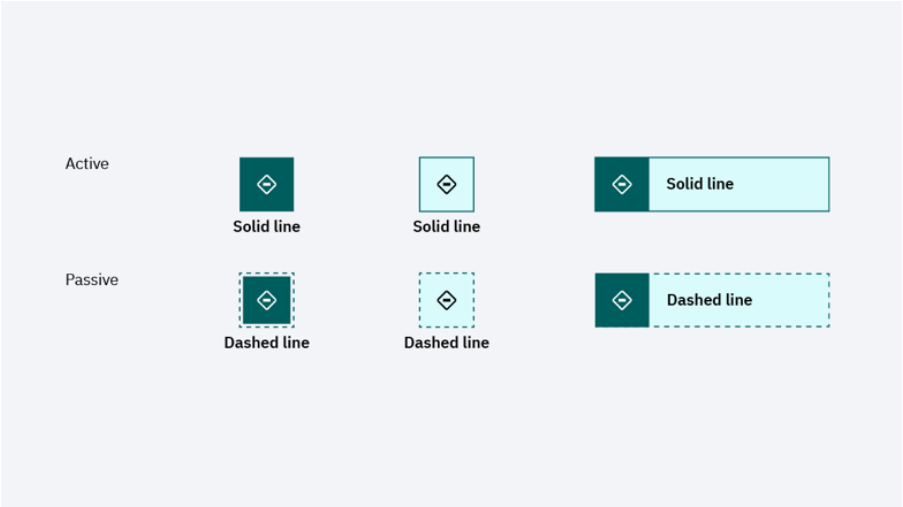
Style

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

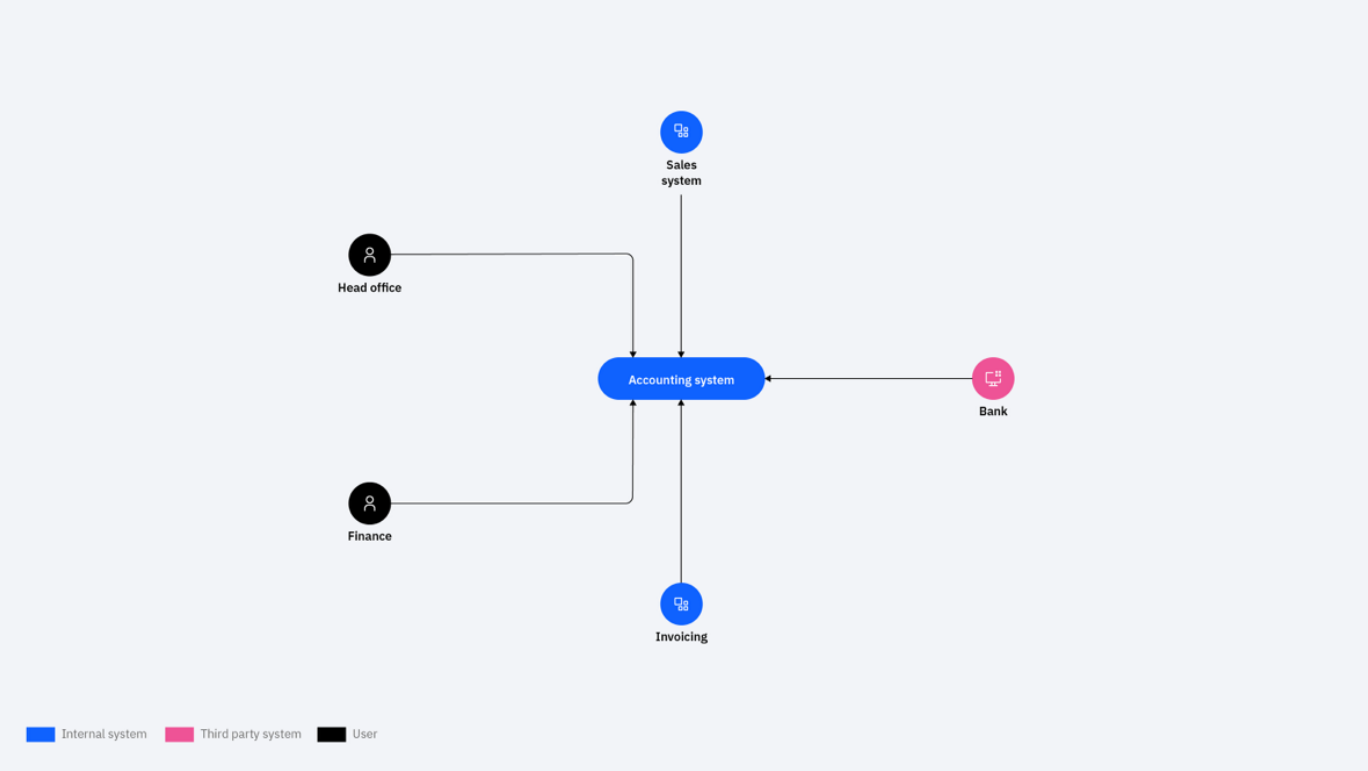
Always include a legend to identify how you have made use of visual techniques, like Colour, Badges, Style.

Architectural views

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

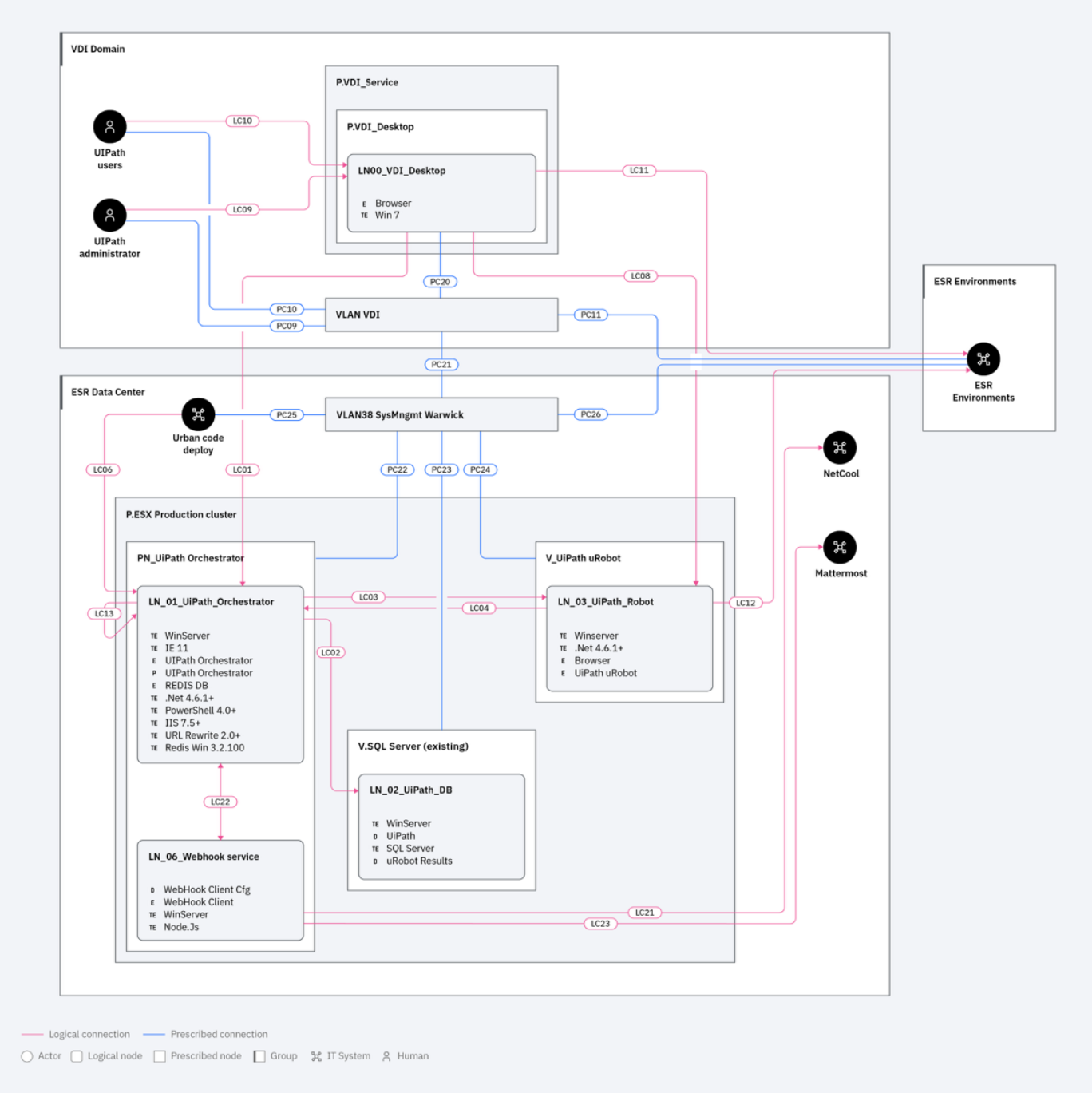
System context diagram

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.



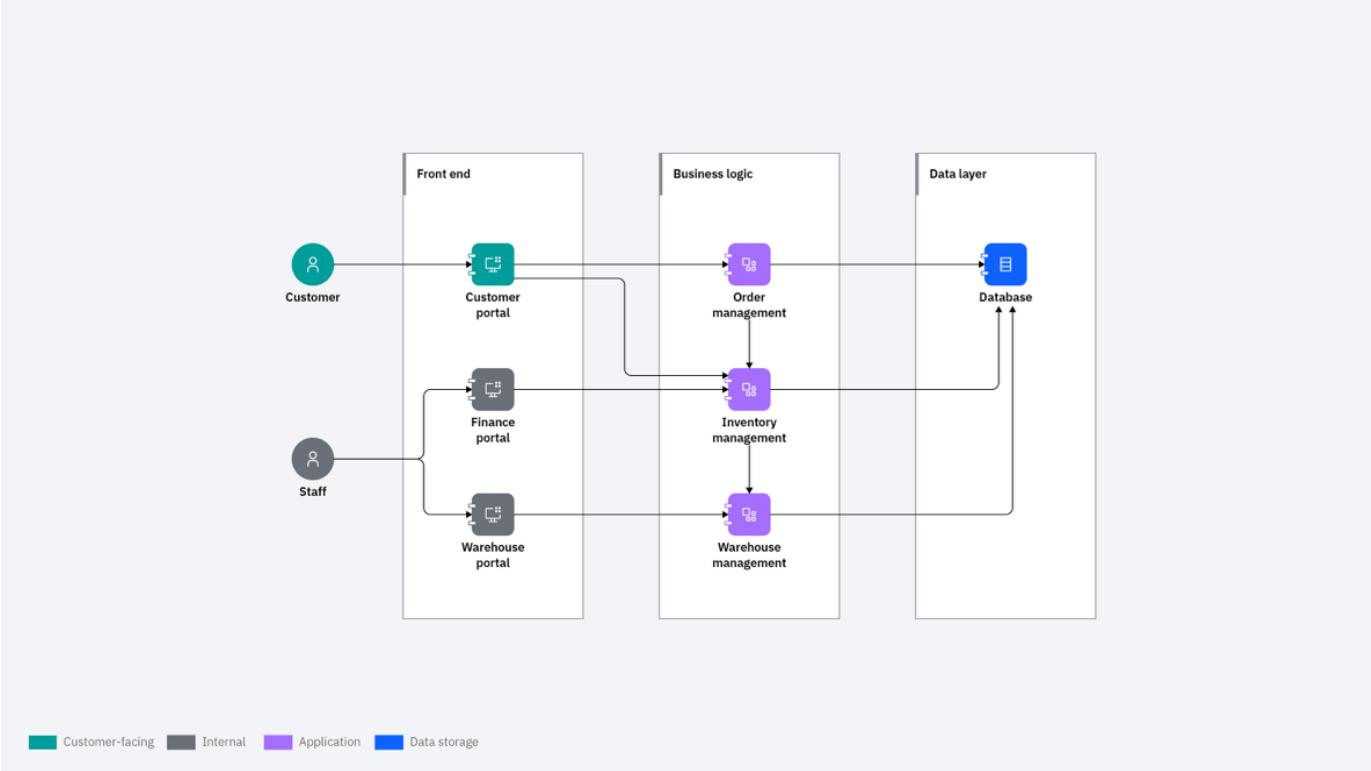
Operational model

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.


VDI Domain
P.VDI_Service
P.VDI_Desktop
LN00_VDI_Desktop
E Browser
TE Win 7
UIPath users
UIPath administrator
LC10
LC09
LC11
LC08
PC20
PC10
PC09
VLAN VDI
PC11
PC21
ESR Environments
ESR Environments
ESR Data Center
Urban code deploy
PC25
VLAN38 SysMngmt Warwick
PC26
LC06
LC01
PC22
PC23
PC24
NetCool
Mattermost
P.ESX Production cluster
PN_UiPath Orchestrator
V_UiPath uRobot
LN_01_UiPath_Orchestrator
TE WinServer
TE IE 11
E UIPath Orchestrator
P UIPath Orchestrator
E REDIS DB
TE .Net 4.6.1+
TE PowerShell 4.0+
TE IIS 7.5+
TE URL Rewrite 2.0+
TE Redis Win 3.2.100
LC13
LC03
LC04
LC12
LC02
LN_03_UiPath_Robot
TE Winserver
TE .Net 4.6.1+
E Browser
E UiPath uRobot
V.SQL Server (existing)
LN_02_UiPath_DB
TE WinServer
D UiPath
TE SQL Server
D uRobot Results
LC22
LN_06_Webhook service
D WebHook Client Cfg
E WebHook Client
TE WinServer
TE Node.Js
LC21
LC23
Logical connection
Prescribed connection
Actor
Logical node
Prescribed node
Group
IT System
Human

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.

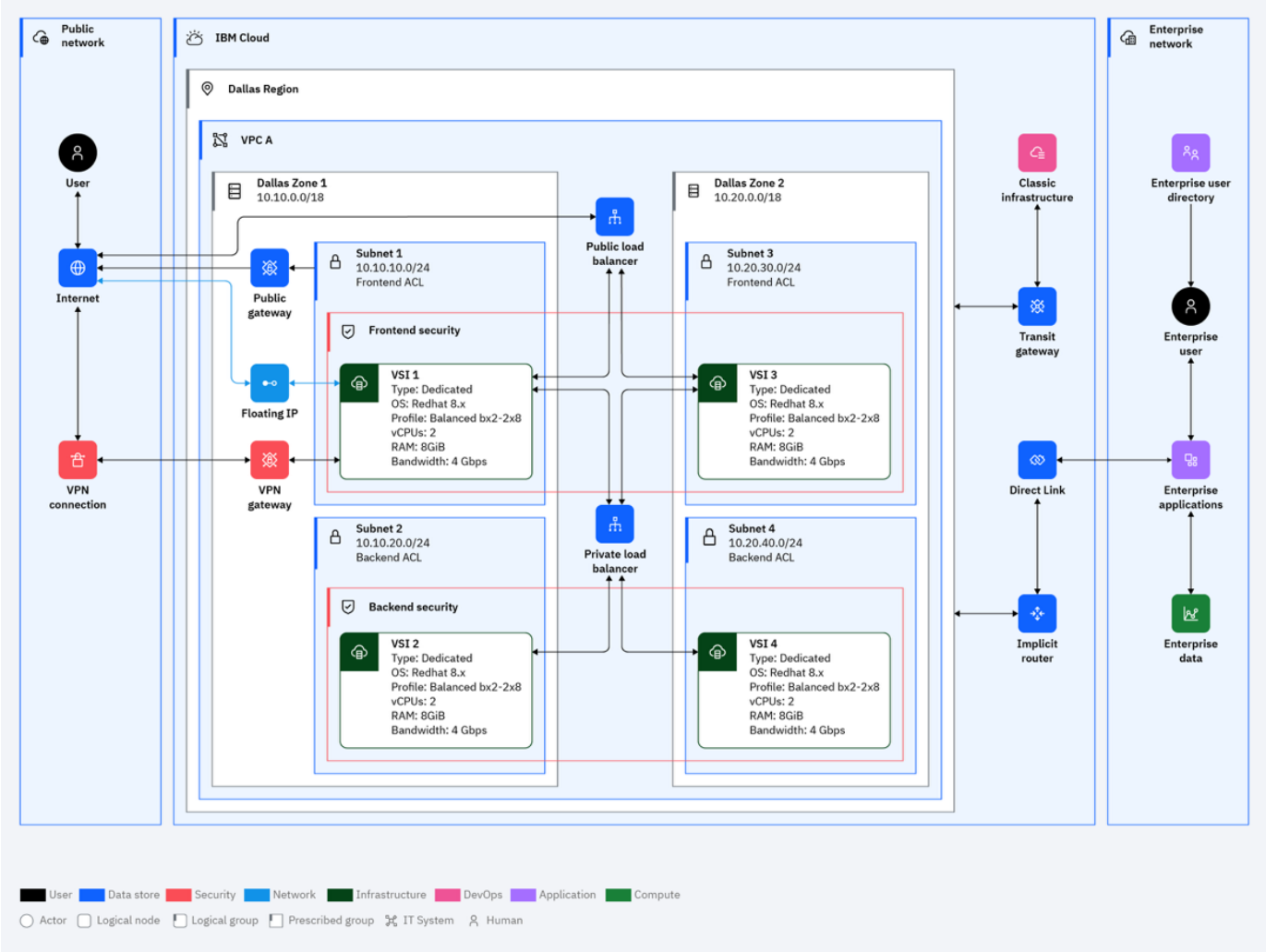


Other related views

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.


Public network
IBM Cloud
Enterprise network
Dallas Region
VPC A
User
Internet
VPN connection
Public gateway
Floating IP
VPN gateway
Dallas Zone 1
10.10.0.0/18
Subnet 1
10.10.10.0/24
Frontend ACL
Frontend security
VSI 1
Type: Dedicated
OS: Redhat 8.x
Profile: Balanced bx2-2x8
vCPUs: 2
RAM: 8GiB
Bandwidth: 4 Gbps
Subnet 2
10.10.20.0/24
Backend ACL
Backend security
VSI 2
Type: Dedicated
OS: Redhat 8.x
Profile: Balanced bx2-2x8
vCPUs: 2
RAM: 8GiB
Bandwidth: 4 Gbps
Public load balancer
Private load balancer
Dallas Zone 2
10.20.0.0/18
Subnet 3
10.20.30.0/24
Frontend ACL
VSI 3
Type: Dedicated
OS: Redhat 8.x
Profile: Balanced bx2-2x8
vCPUs: 2
RAM: 8GiB
Bandwidth: 4 Gbps
Subnet 4
10.20.40.0/24
Backend ACL
VSI 4
Type: Dedicated
OS: Redhat 8.x
Profile: Balanced bx2-2x8
vCPUs: 2
RAM: 8GiB
Bandwidth: 4 Gbps
Classic infrastructure
Transit gateway
Direct Link
Implicit router
Enterprise user directory
Enterprise user
Enterprise applications
Enterprise data
User
Data store
Security
Network
Infrastructure
DevOps
Application
Compute
Actor
Logical node
Logical group
Prescribed group
IT System
Human

Resources

Tools and kits

Cognitive Architect

Diagrams.net (ex draw.io)

Adobe Illustrator kit

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