**C4 Diagramming Model**

C4 model documents the architecture of a software system, by showing multiple points of view that explain the decomposition of a system into containers and components, the relationship between these elements, and, where appropriate, the relation with its users.

**C4 Diagramming Model**

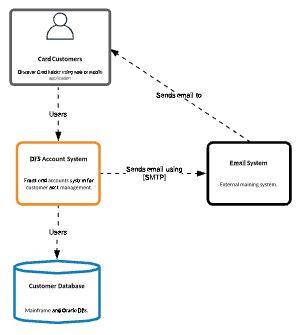
C4 model documents the architecture of a software system, by showing multiple points of view that explain the decomposition of a system into containers and components, the relationship between these elements, and, where appropriate, the relation with its users.

Within ABC, we are following the C4 Diagramming Model, but we changed the labels for each level to "System Context (Level 1)", "Components (Level 2)", "Sub-Components (Level 3)" and "Code (Level 4)". Within C4, they use "Context (Level 1)", "Containers (Level 2)", "Components (Level 3)" and "Code (Level 4)". We determined that the use of "Containers" is confusing, since it has different meanings within ABC.

The viewpoints are organized according to their hierarchical level:

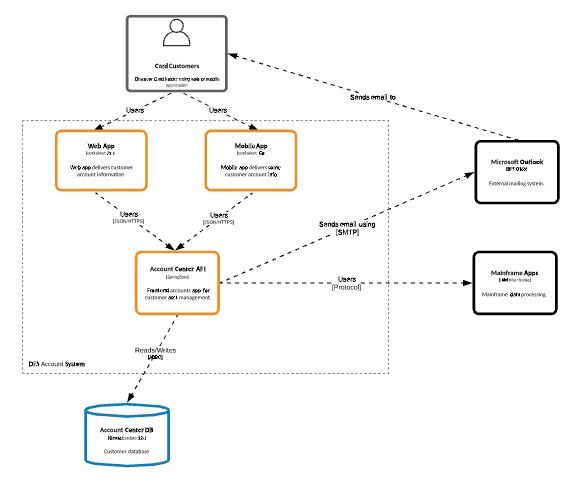
**Context diagrams (level 1):**

Level 1, a system context diagram, shows the software system you are building and how it fits into the world in terms of the people who use it and the other software systems it interacts with.



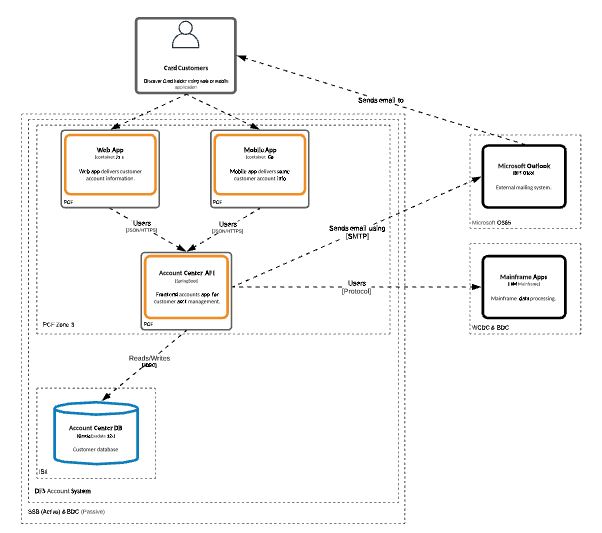
**Component diagrams (level 2):**

Level 2, a component diagram, zooms into the software system, and shows the components (applications, data stores, microservices, etc.) that make up that software system. Technology decisions are also a key part of this diagram.



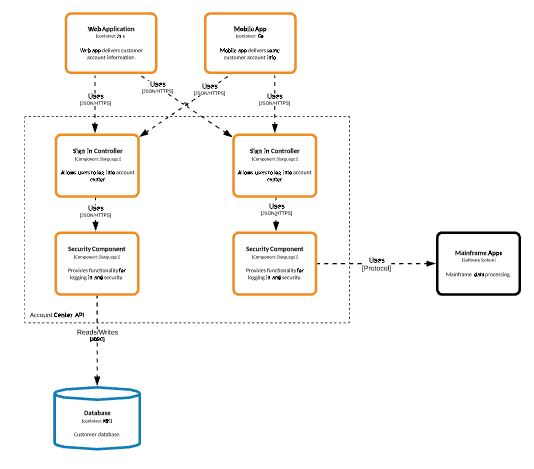
**Physical diagram:**

A physical diagram shows each physical/virtual device, its interconnectivity, and its placement within the DC/Cloud environment and/or security zone.



**Sub-Component diagrams (level 3 - optional):**

Level 3, a sub-component diagram, is optional but recommended. It zooms into an individual components to show the sub-components inside it. These sub-components should map to real abstractions (e.g., a grouping of code) in your codebase.



**Code diagrams (level 4):**

Provide additional details about the design of the architectural elements that can be mapped to code. C4 model relies at this level on existing notations such as Unified Modelling Language (UML), Entity Relation Diagrams (ERD) or diagrams generated by Integrated Development Environments (IDE).

For level 1 to 3, the C4 model uses 5 basic diagramming elements: persons, software systems, components, sub-components and relationships. The technique is not prescriptive for the layout, shape, colour and style of these elements. Instead, the C4 model recommends using simple diagrams based on nested boxes in order to facilitate interactive collaborative drawing. The technique also promotes good modelling practices such as providing a title and legend on every diagram, and clear unambiguous labelling in order to facilitate the understanding by the intended audience.

C4 model facilitates collaborative visual architecting and evolutionary architecture in the context of agile teams where more formal documentation methods and up-front architectural design are not desired.

**Templates/Stencils/Lucid:**

Lucid Chart is the current standard drawing tool for ABC. To get access to this web based app, request via ServiceNow.

Diagram Templates are currently being added to Lucid. Sharing is pending Lucid Product Team.

Stencils are available in the Lucid Shape Library, search for "C4 Model Library"