C4 Models

- Need a language for communicating software architecture that is:
  - consistent
  - expressive enough to represent physical and logical architecture (tiers and layers)
  - describe the system at different levels of abstraction
  - easy to use in a design discussion
- Abstractions
  - Person
  - Software System - may be ours or one ours interacts with
  - Container
    - something that hosts code or data - needs to be running for our system to work
    - a separately runnable/deployable unit
  - Component - grouping of related functionality encapsulated behind a well-defined interface
- Diagrams
  - Context
    - describes the connections between our system and users or other systems
    - no details about the internal aspects of our systems
    - focus on the environment the system interacts with
    - relationships between the system and the other entities describe how they interact
  - Container
    - Shows the system's containers inside the system box
    - Has connections to the other entities in the system's context
    - For each container, specifies:
      - name
      - technology
      - description
    - Relationships describe how containers are used and use each other
  - Component
    - Shows the portion of the system that is relevant to one container
    - Decomposes a container into its components
    - For each component, specifies
      - name
      - technology
      - description
  - Code
    - one code diagram for each component
    - How it is implemented

    - UML class or entity relationship diagrams

- Other suggested diagrams
    - System Landscape diagram
        - can show how several software systems work together to provide a service
        - essentially one layer above a context diagram
    - Dynamic diagram
        - Shows how elements collaborate to accomplish one task
        - can be for an enterprise, a software system, or a container
    - Deployment diagram
        - shows how containers are mapped to infrastructure