

# Design

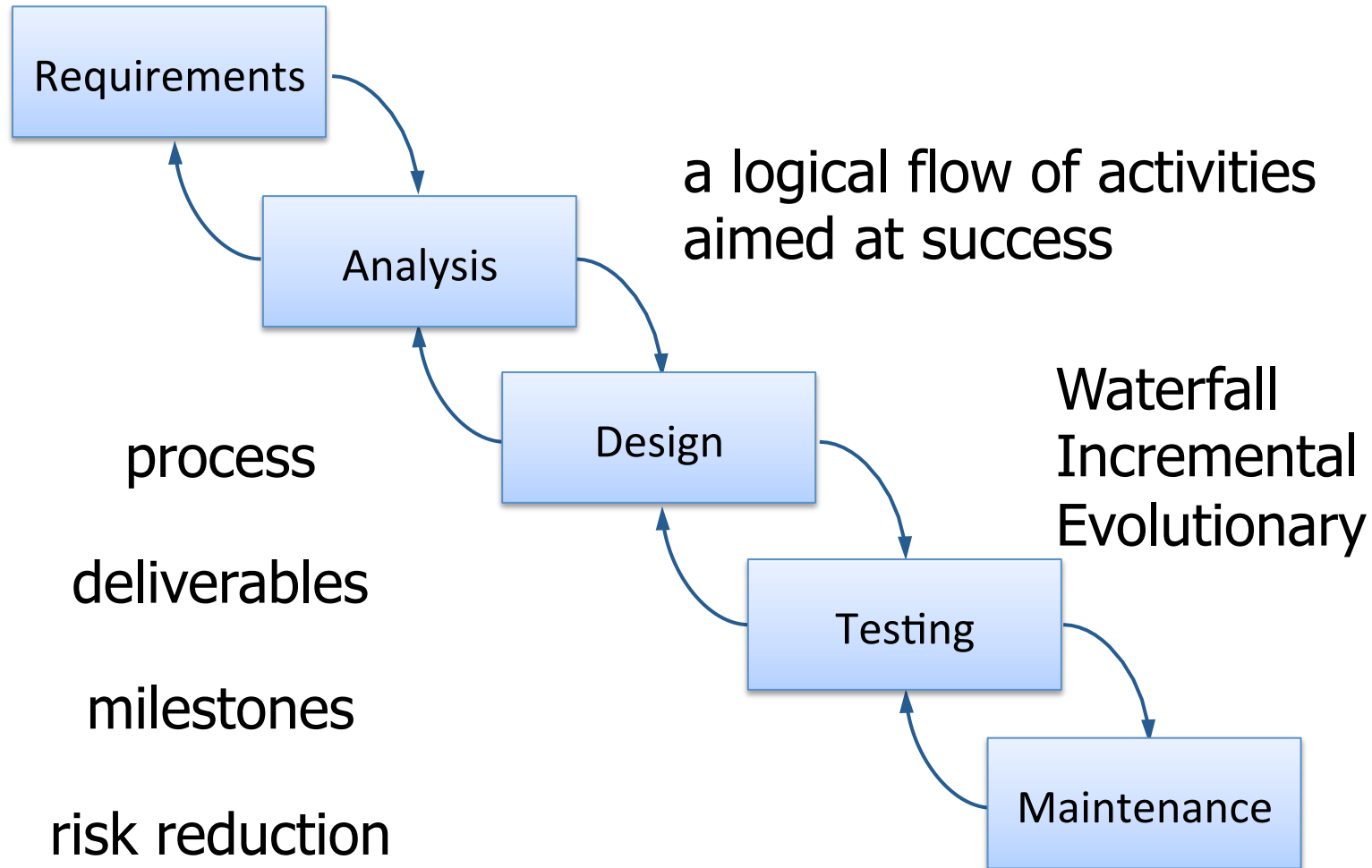
EEE/GEF 455/457



Royal Military College of Canada  
Electrical and Computer Engineering

# Recall

- Engineering Process



# Requirement's Review

- Characteristics of Good Requirement?
  - What ( **not how** )
  - A Single Idea
  - Measurable
  - Necessary
  - Concise
  - Implementation Free
  - Attainable
  - Complete
  - Consistent
  - Unambiguous
  - Verifiable

# Design

- An ability to design solutions for complex, open-ended engineering problems and to design systems, components or processes that meet specified needs with appropriate attention to health and safety risks, applicable standards, and economic, environmental, cultural and societal considerations.
  - The Canadian Engineering Accreditation Board

# Why Design?

- Prevent waste
- Get customer approval
- Improve Chance of success
- Assist with planning
- Evaluate options
- Reduce risks
- Manage Complexity!

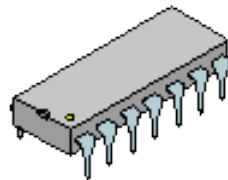
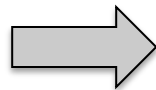
# Managing Complexity

Design complexity is managed via three fundamental techniques:

- abstraction
- decomposition
- hierarchy

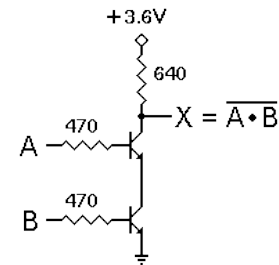
# Managing Complexity

- Abstraction
  - overlook the unnecessary details and focus only upon the essential aspects relevant to the current view of the system
  - an inherent human ability (fortunately for us!)



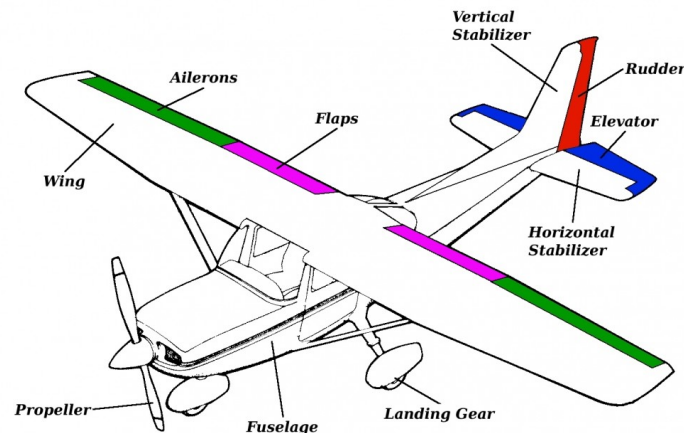
Meaning?

<u>A</u>	<u>B</u>	<u>(AB)'</u>
0	0	1
0	1	1
1	0	1
1	1	0



# Managing Complexity

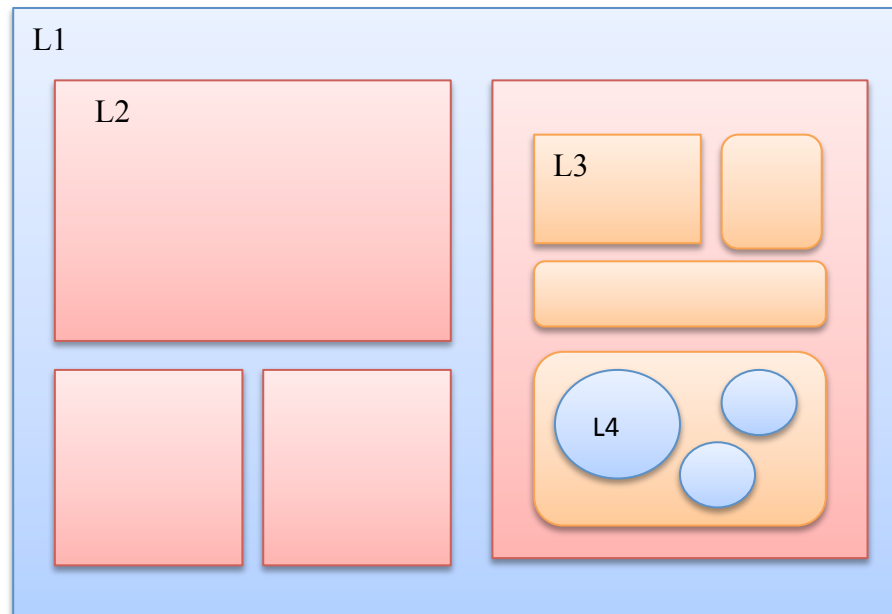
- Decomposition
  - break a problem into manageable components
  - distribute effort (and schedule) across the pieces
  - create well defined building blocks
  - separate issues (coupling vs cohesion)



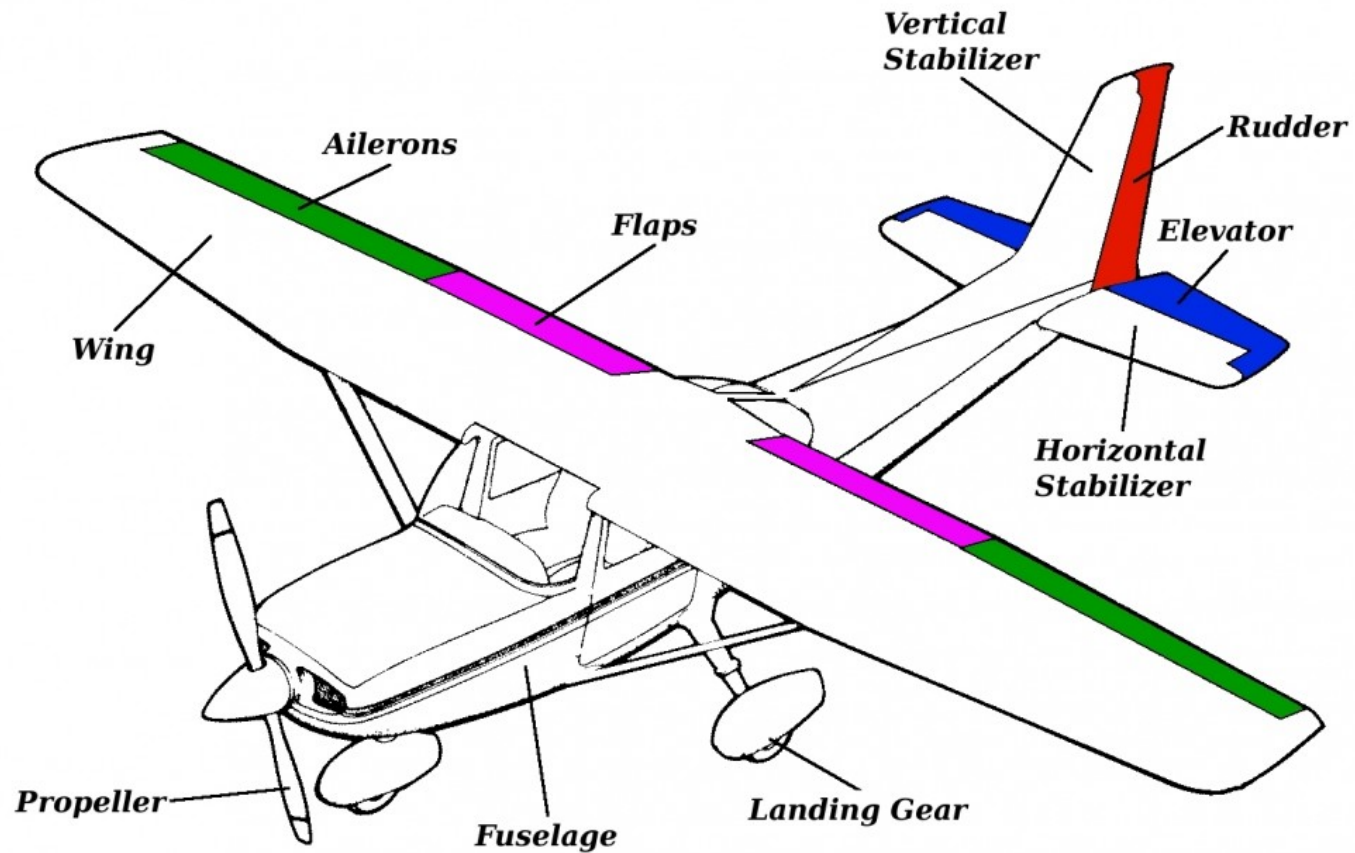


# Managing Complexity

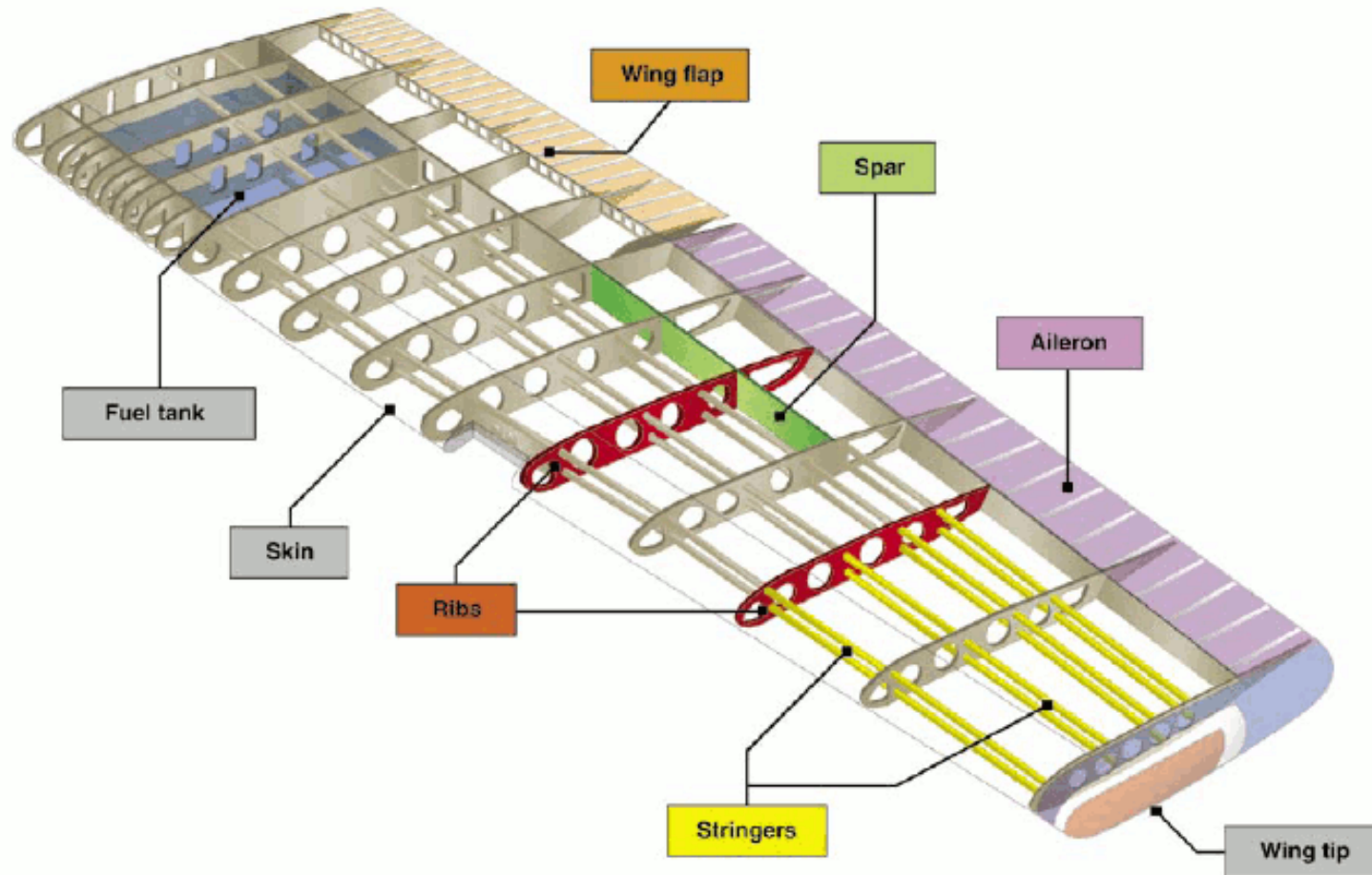
- Hierarchy
  - build views of the system through relationships of the components
  - layer your decomposition from higher levels (L1, L2) of abstraction to lower levels (L3, L4, ...)



# Hierarchy Example



# Hierarchy Example



# Hierarchy Example

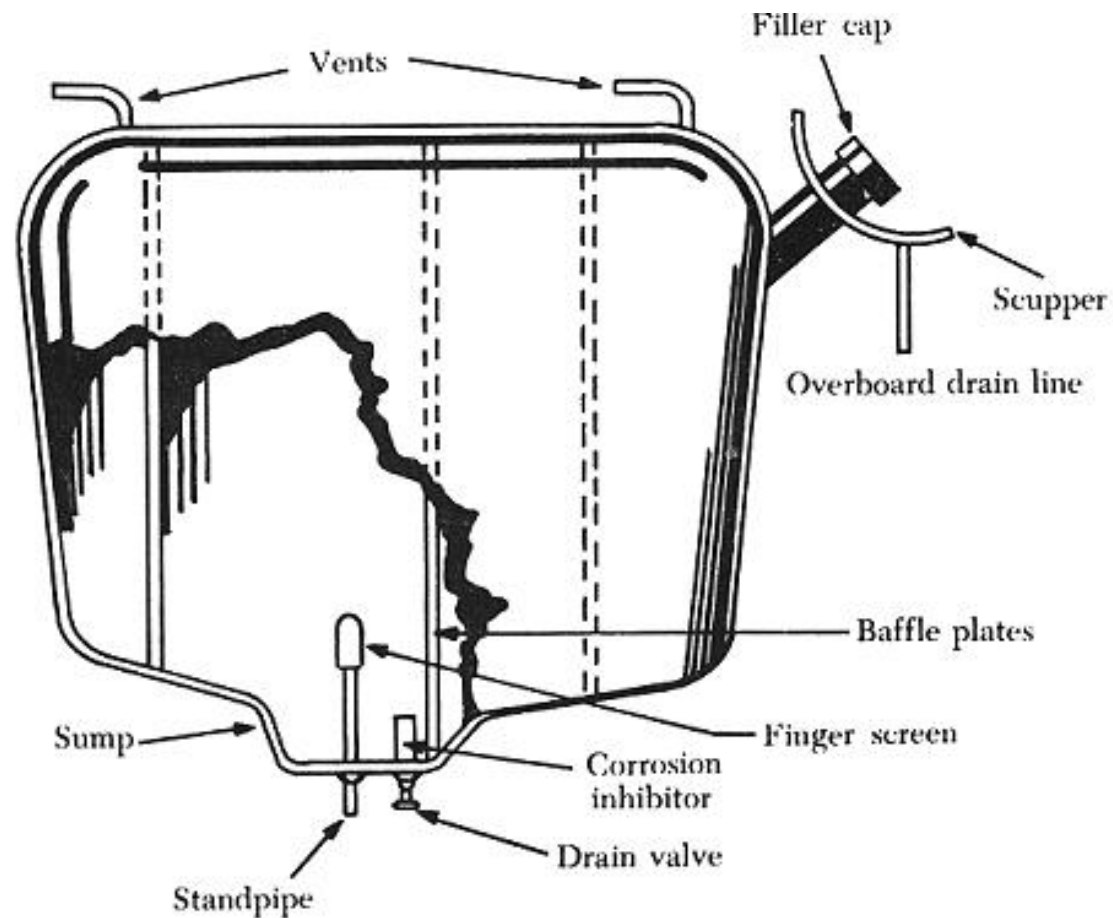
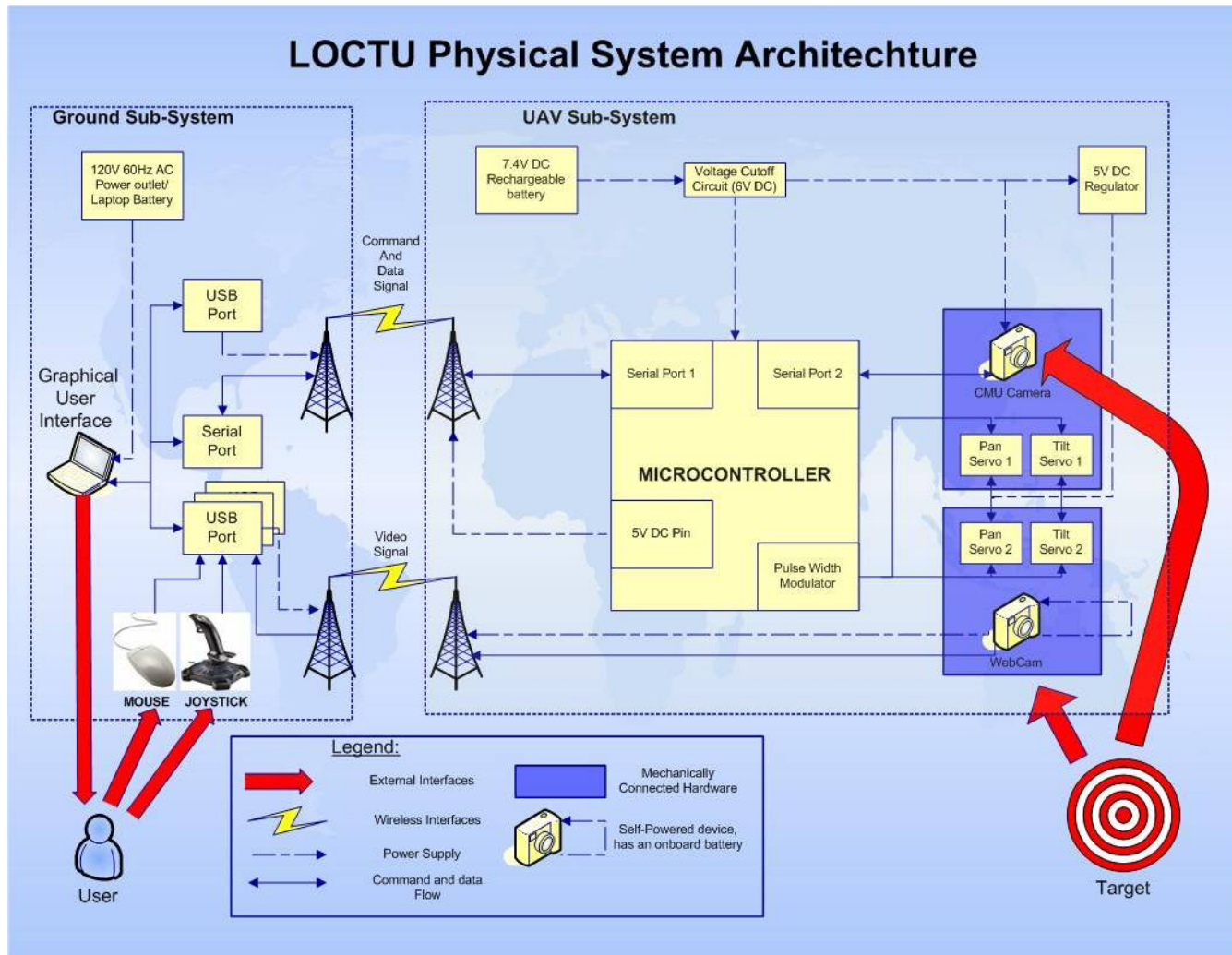


FIGURE 4-8. A typical metal fuel tank.

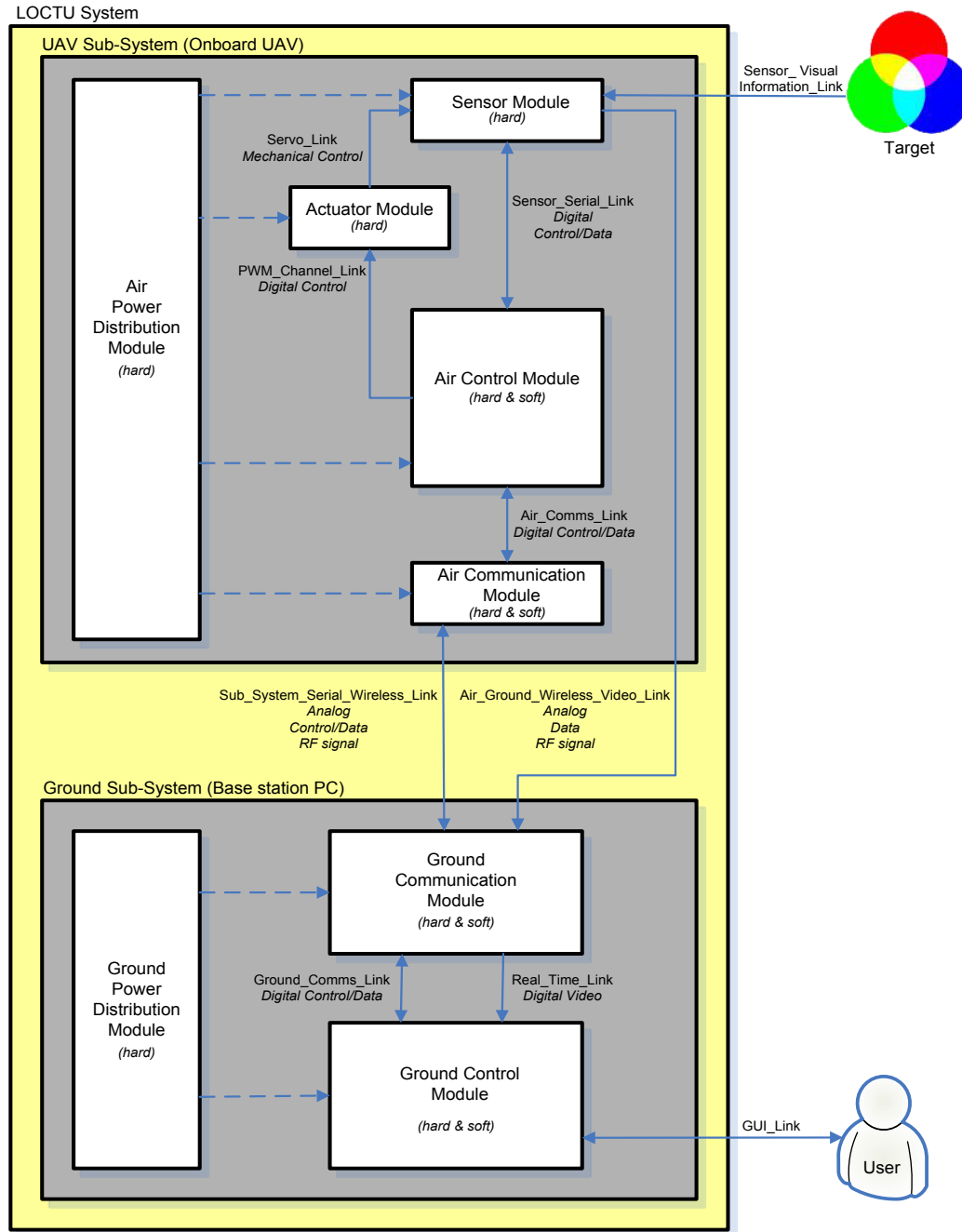
# Architectural (preliminary) Design

- Represent:
  - an abstraction of the system as a whole
  - the second layer of decomposition
  - the functional (block diagram) view
- Ensure:
  - that you know external interfaces
  - that each major component is cohesive
  - that the internal interfaces are well defined

# Architectural (preliminary) Design



# Functional Diagram

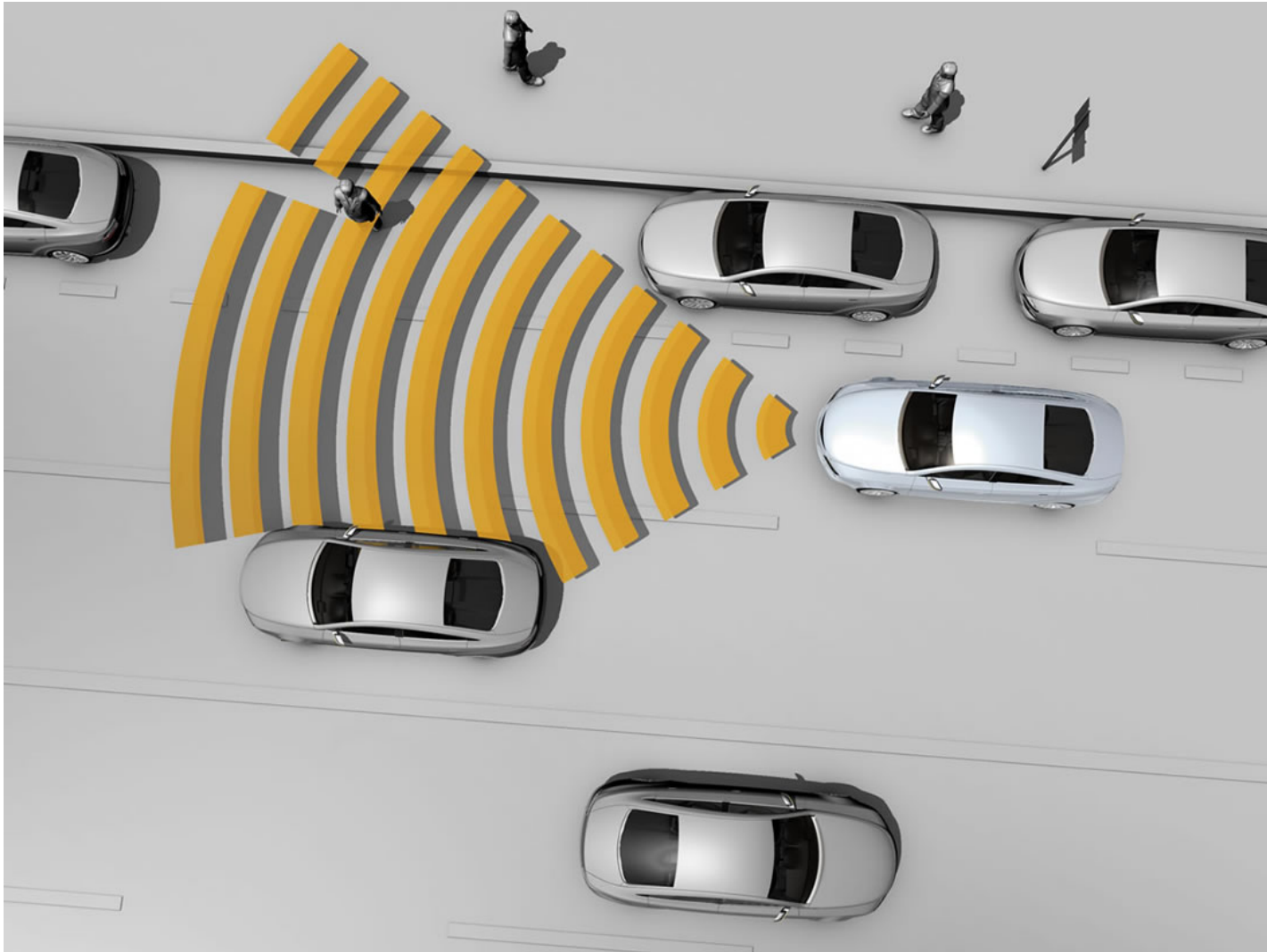


# Preliminary Design Specification

**See your SOW / DID-04**



# Exercise



# Question?

