National Cyber Security Centre

Scoping assurance activities with seL4

Gage – The National Cyber Security Centre

Who are the NCSC?

Who am I?

- Assurance Lead within the NCSC
- Expertise in Cryptography and Software Assurance
- The NCSC seL4 Assurance Expert

Why do we need product assurance?

- If the product doesn’t function correctly, people won’t want it
  - Reputational risk
  - Financial loss
- To give us confidence that the product is safe or secure
- To mitigate the risk of a product failure impacting safety or security
  - Plane falls out the sky
  - Encryption fails and data is compromised

What do I mean by scoping assurance activities?

- Assurance activities build evidence to demonstrate that the product meets its security requirements
- It’s about doing enough, and enough depends on the context
- With seL4 in mind, the focus is on software assurance

What do I think is important to be confident in the software of a security product?

1. The software is free from all known bugs and vulnerabilities

2. The security-critical software does what it is supposed to do

3. The security-critical software doesn't do what it is not supposed to do

So what?

How can seL4 help with these?

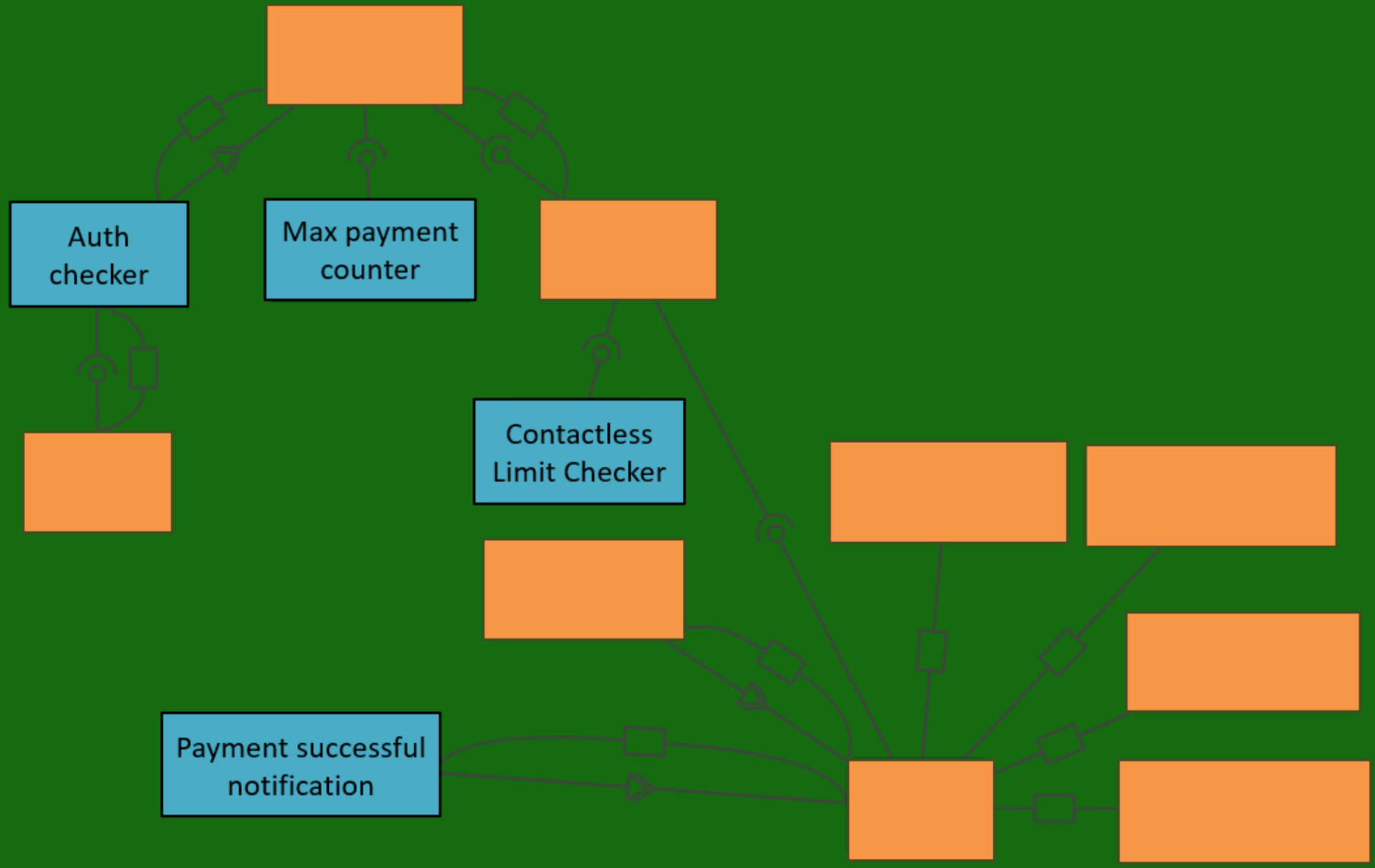
National Cyber Security Centre

Context: A contactless card payment terminal

Payment terminal security critical functions

- Authentication check on the card (challenge-response)
- Contactless payment limit check
- Maximum number of payments before enforcing PIN entry
- Payment success or fail notification
- …

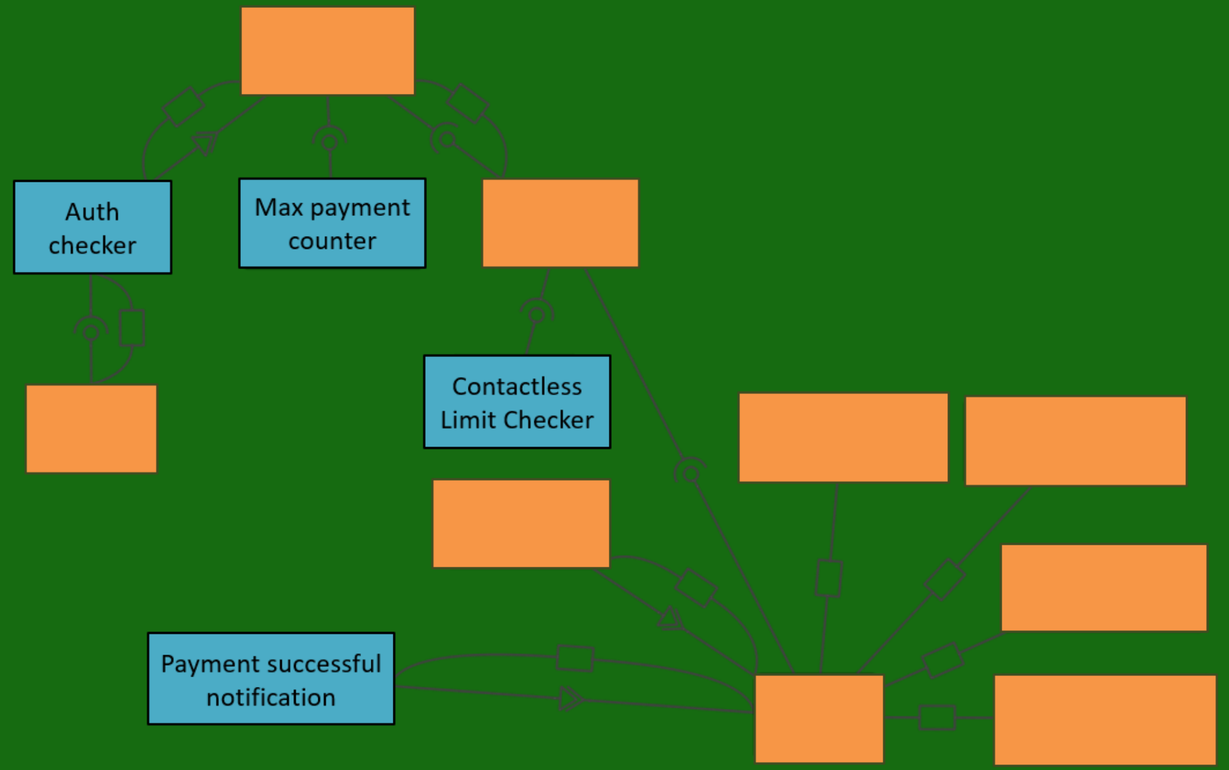
National Cyber Security Centre

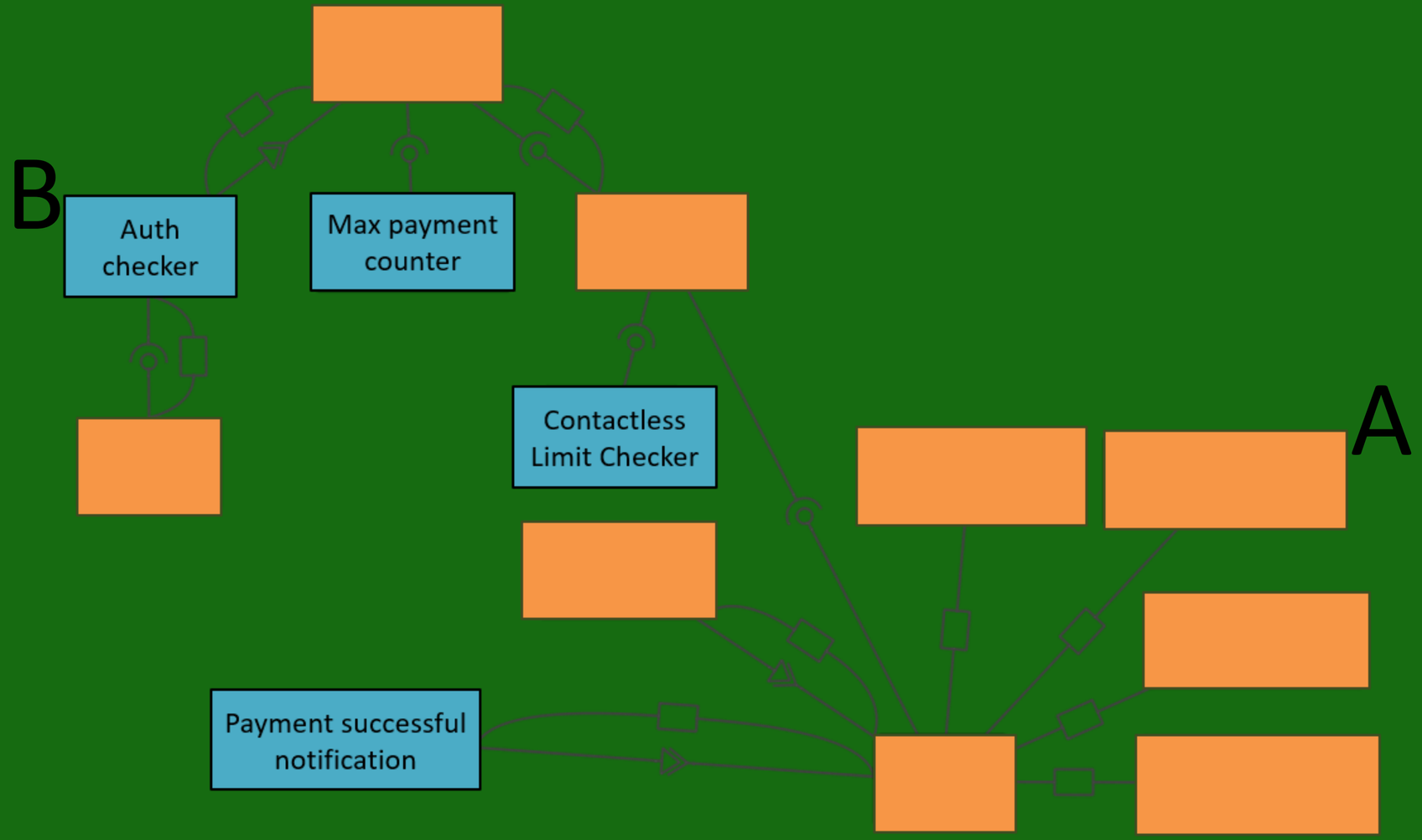


https://docs.sel4.systems/projects/camkes/visual-camkes/

1) The software is free from all known bugs and vulnerabilities

- Product developers writing the same user space code
- … producing the same bugs and vulnerabilities





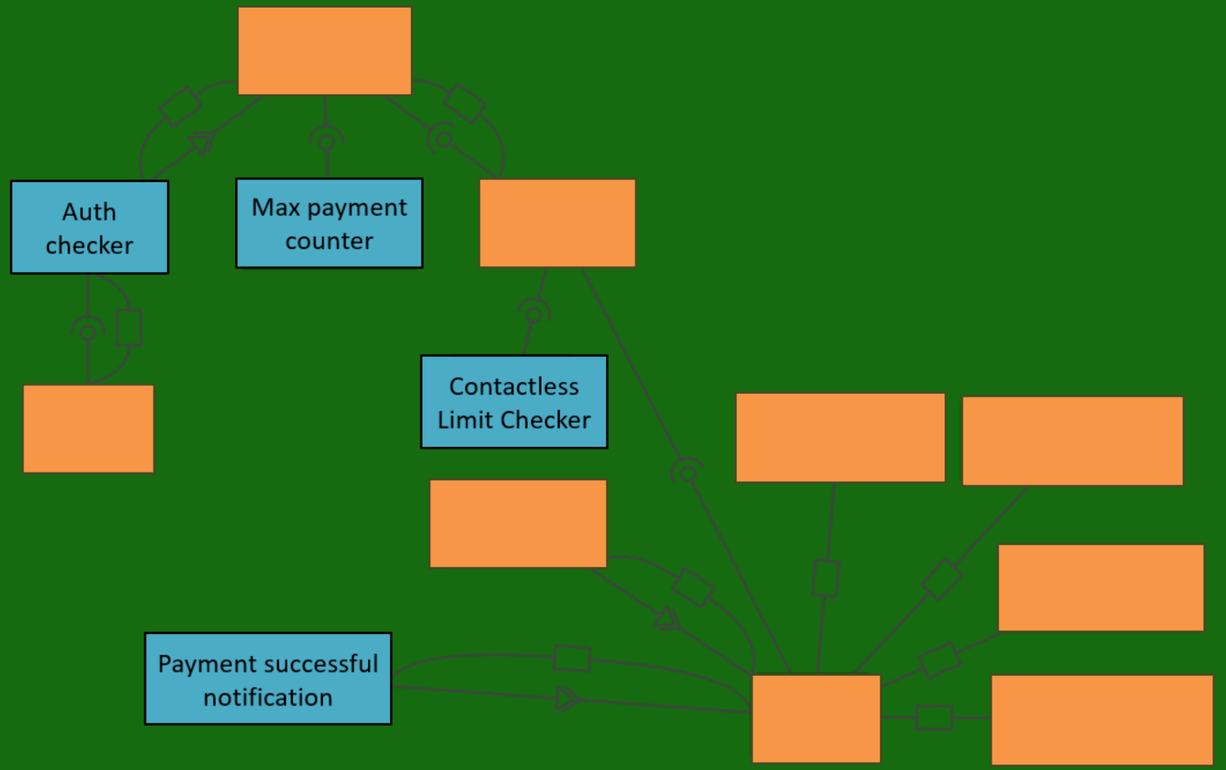
Does using seL4 make it easier to achieve these objectives?

1. The software is free from all known bugs and vulnerabilities ✓
2. The security-critical software does what it is supposed to do
3. The security-critical software doesn't do what it is not supposed to do

National Cyber Security Centre

2) The security-critical software does what it is supposed to do

- We know precisely where the security-critical functions are
- Component separation provides better resilience



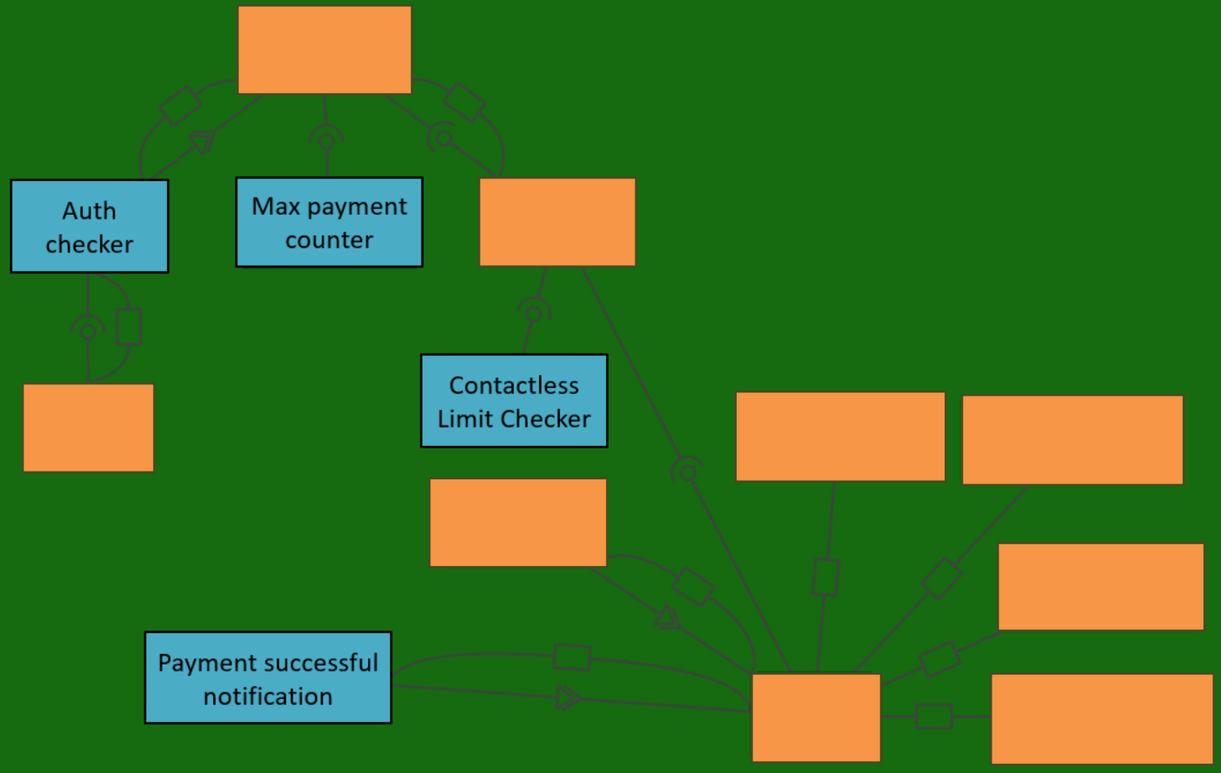
Does using seL4 make it easier to achieve these objectives?

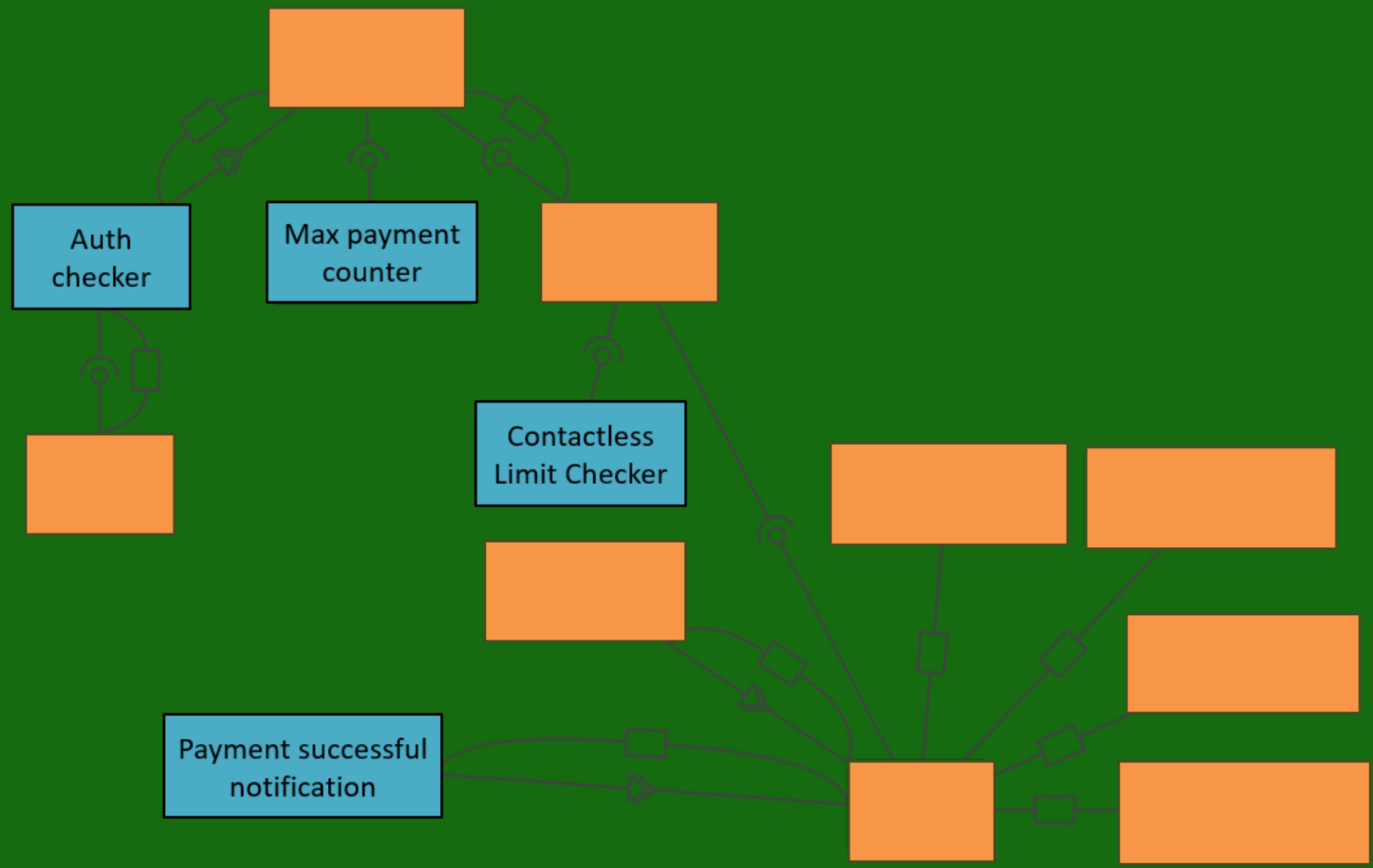
1. The software is free from all known bugs and vulnerabilities ✓
2. The security-critical software does what it is supposed to do –
3. The security-critical software doesn't do what it is not supposed to do

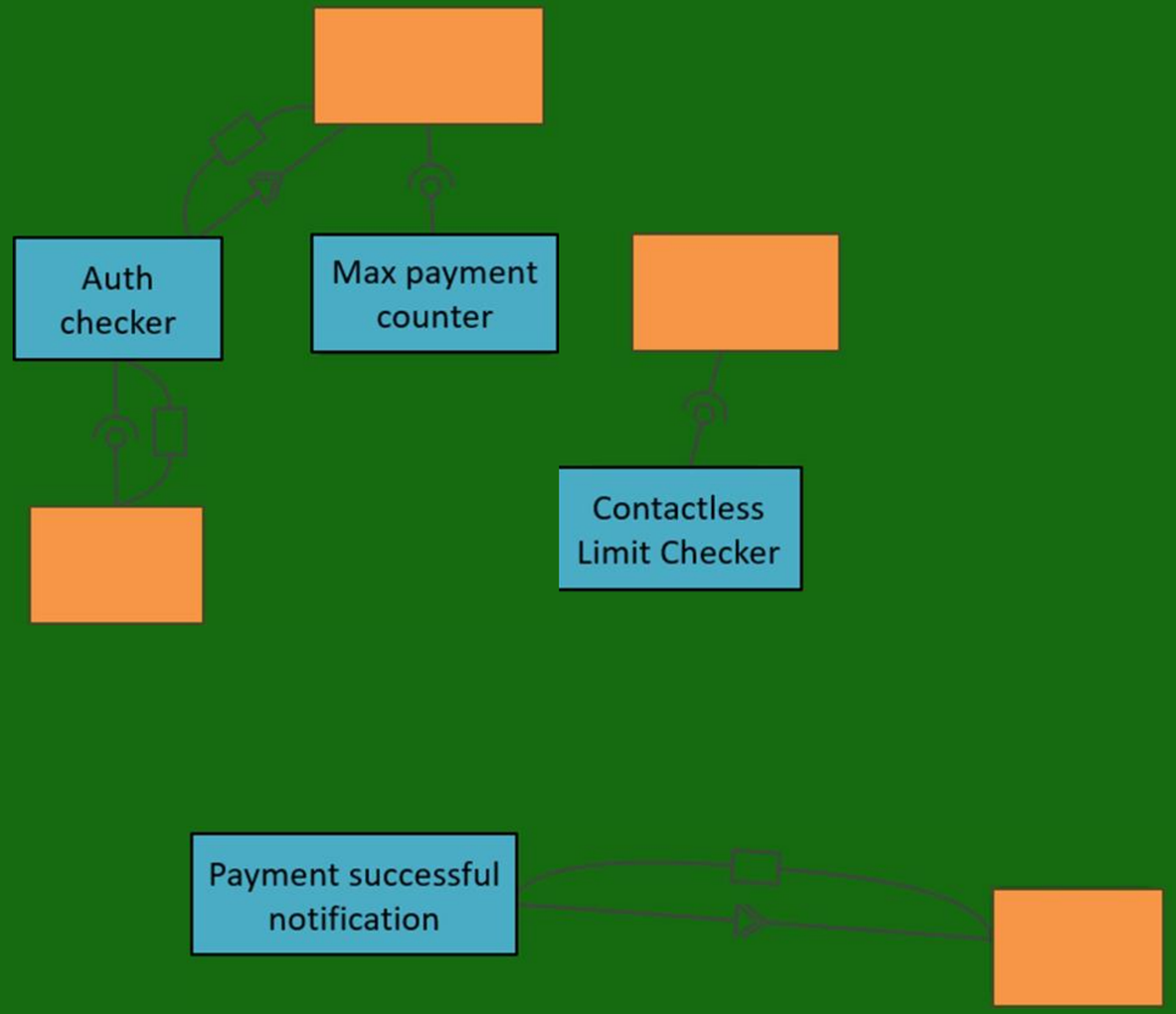
National Cyber Security Centre

3) The security-critical software doesn’t do what it is not supposed to do

- We know where the security-critical things we don’t want to happen could be

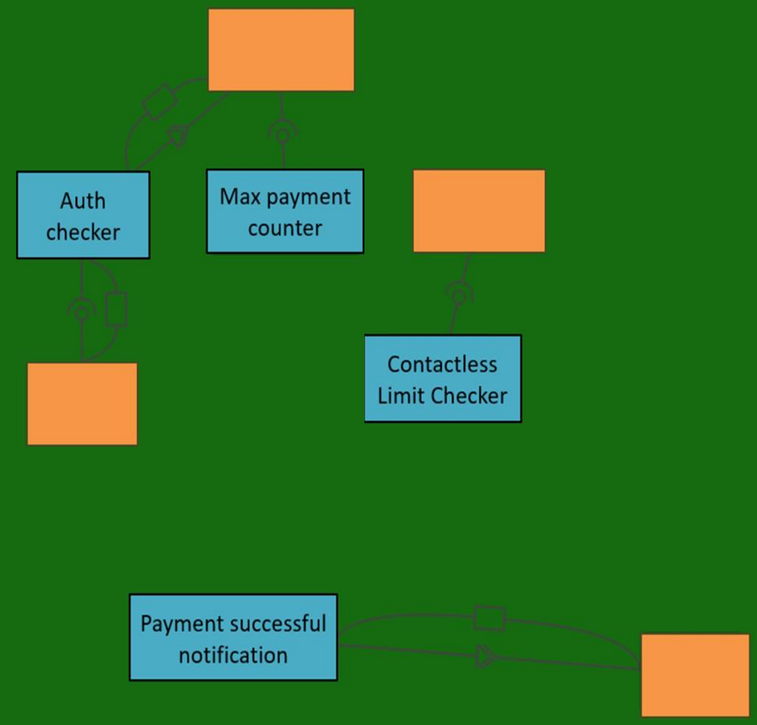






Plan of action:

1. Start at the interface between blue and orange
2. Identify what functionality exists at that interface
3. Understand the blue-side parameters required for that functionality
4. Validate those parameters to be confident of their provenance and correctness



Does using seL4 make it easier to achieve these objectives?

1. The software is free from all known bugs and vulnerabilities 
2. The security-critical software does what it is supposed to do 
3. The security-critical software doesn’t do what it is not supposed to do 

National Cyber Security Centre

The assurance spectrum

The assurance spectrum

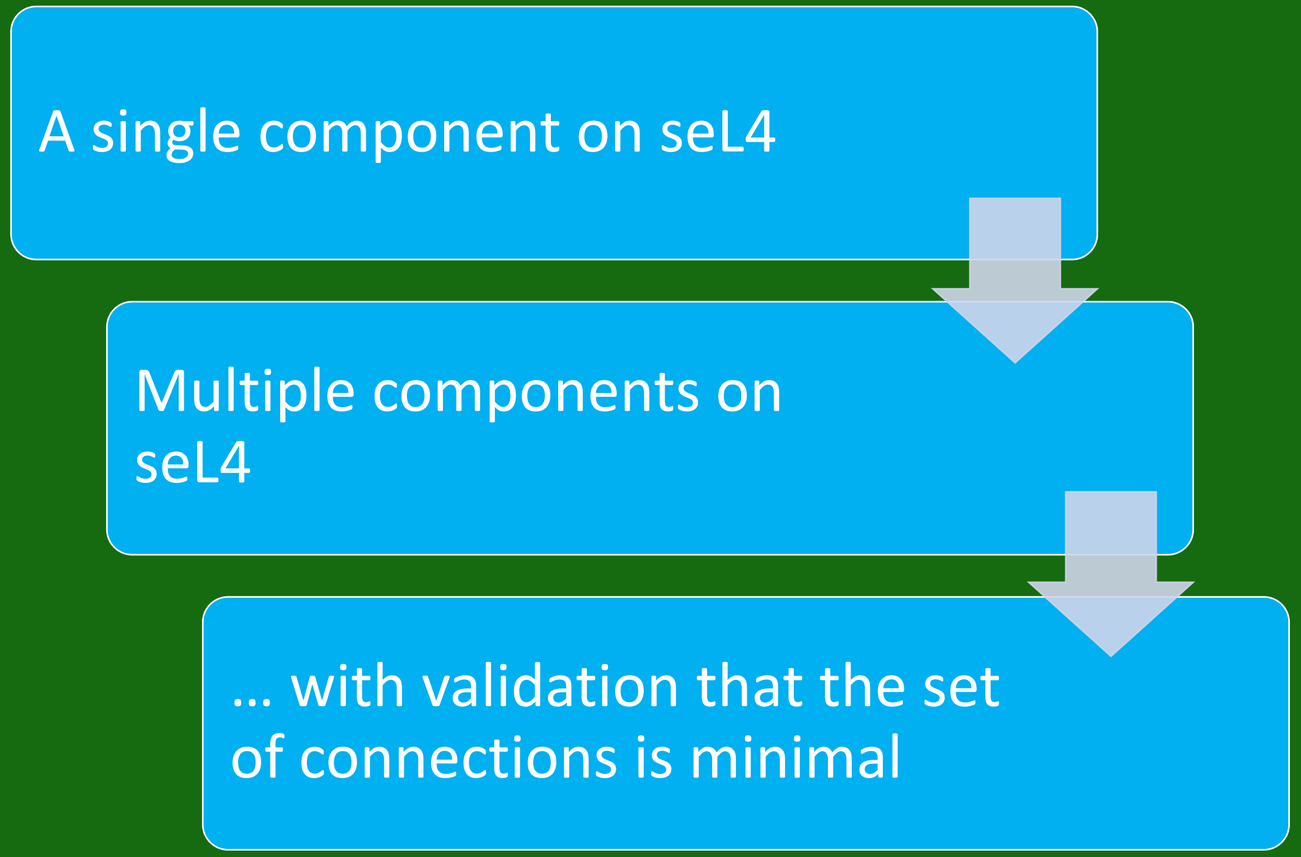
A single component on seL4

The assurance spectrum

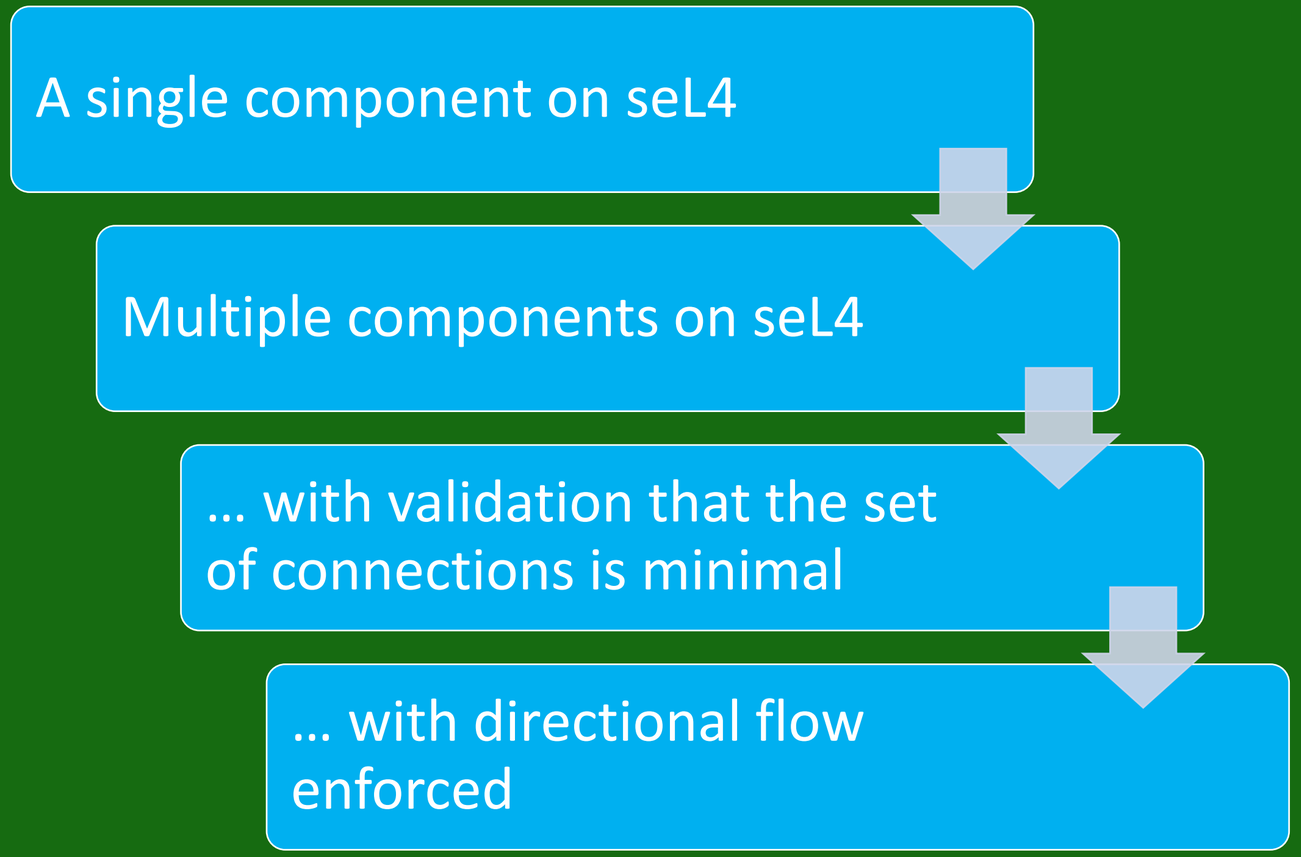
A single component on seL4

Multiple components on seL4

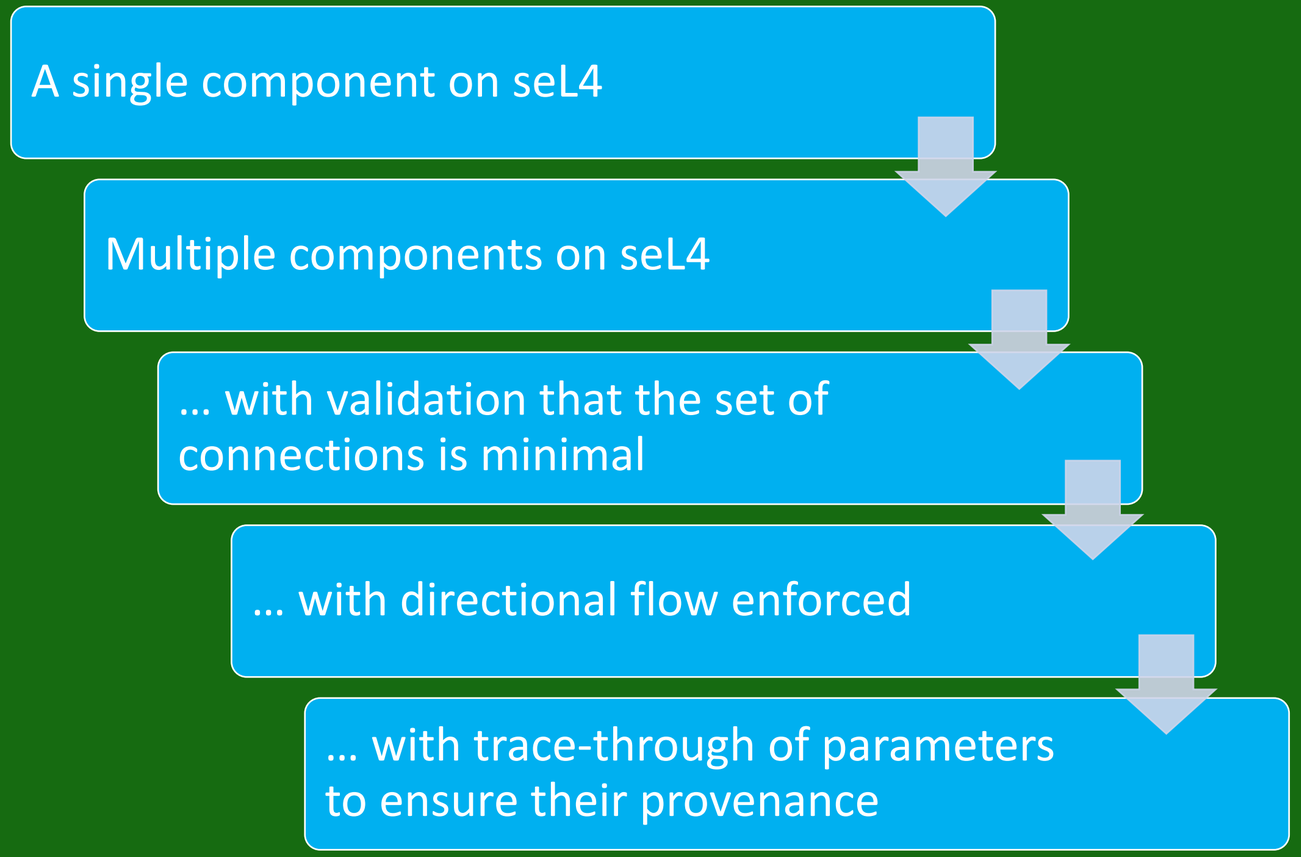
The assurance spectrum



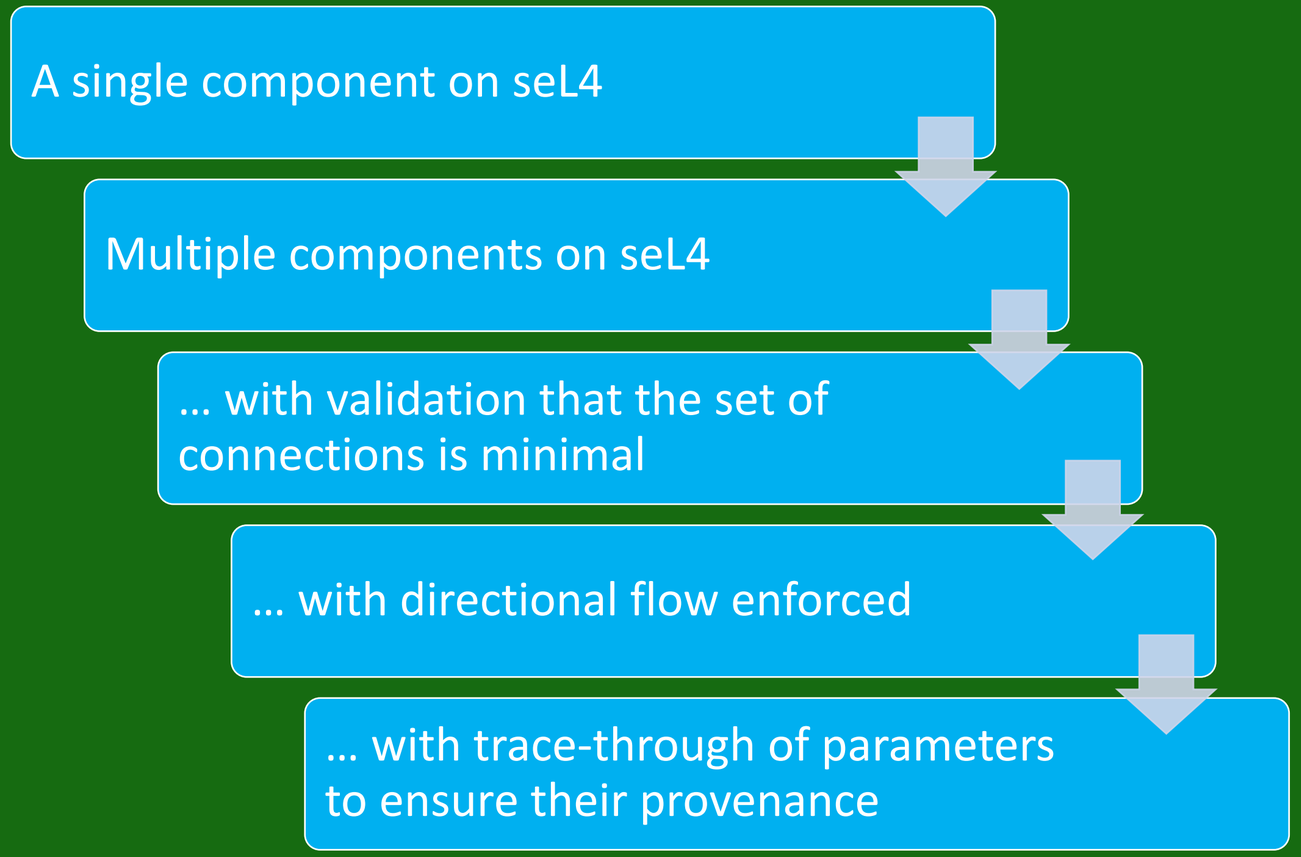
The assurance spectrum



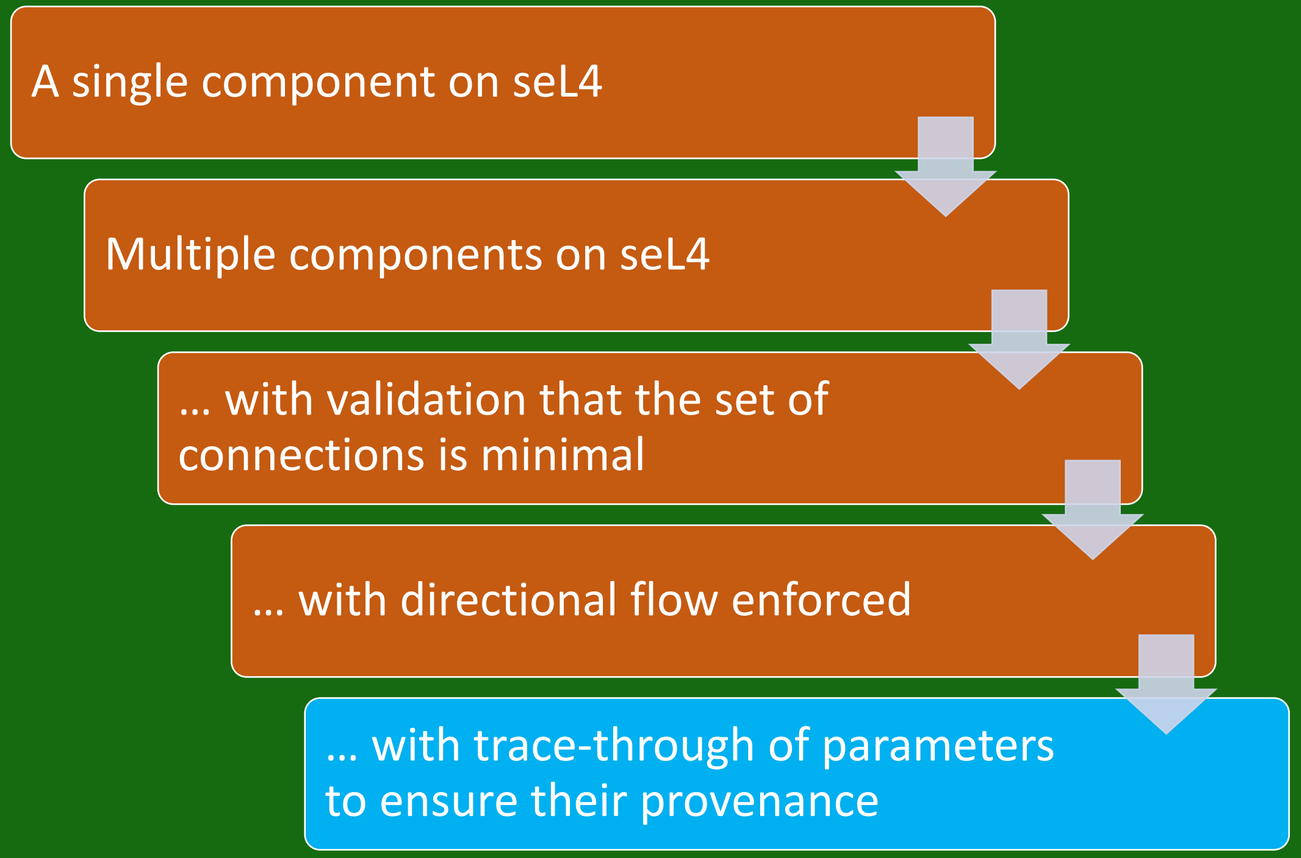
The assurance spectrum



Through life assurance



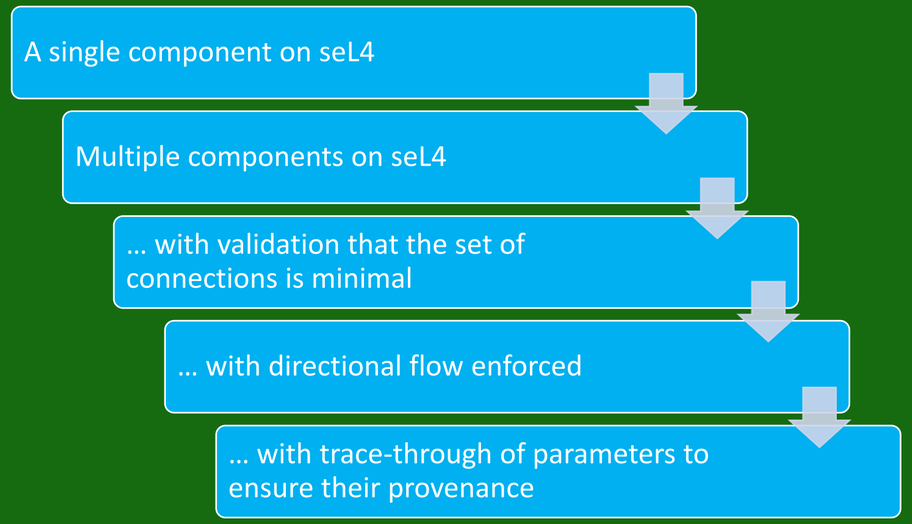
Through life assurance



Core Platform?

Using unproven hardware / features?

- Development and verification is ongoing:
  - MCS Kernel Extension
  - AARCH64 port



1. Developer behaviour
2. Mapping the design to the implementation

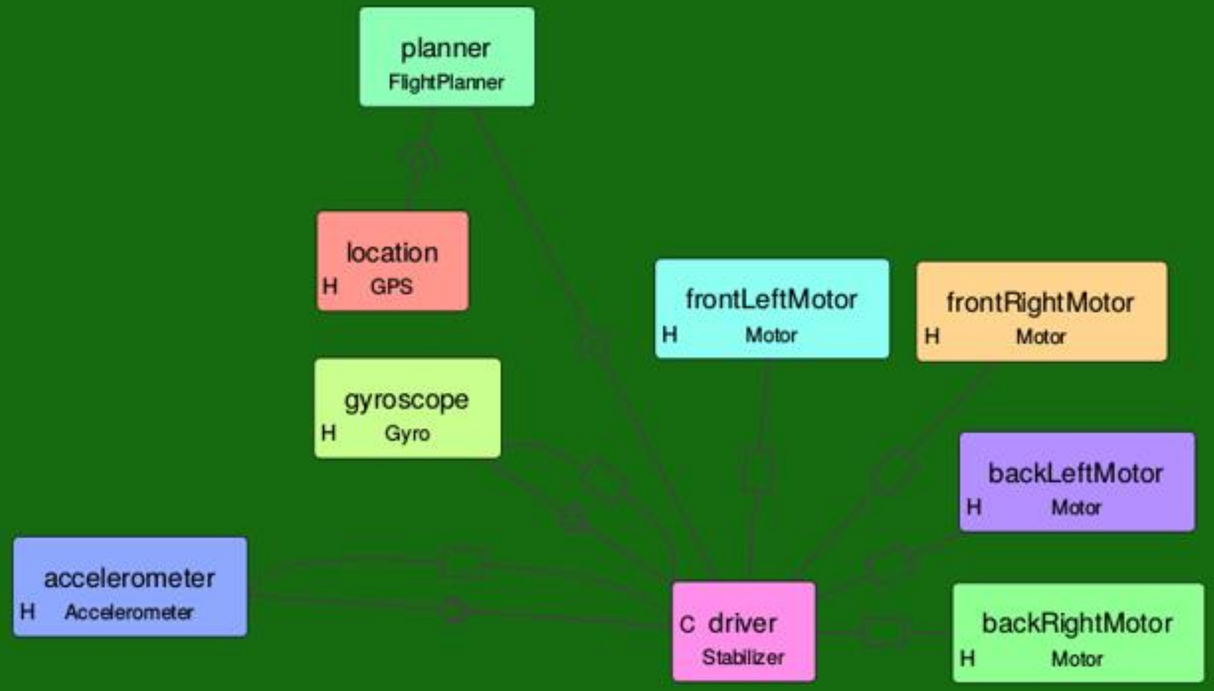
Driving the right developer behaviours

- seL4 installs a modular style of thinking (or developing)
- Improved source code maintenance
- WARNING: Make sure things don’t fall in the gaps

Mapping design to implementation

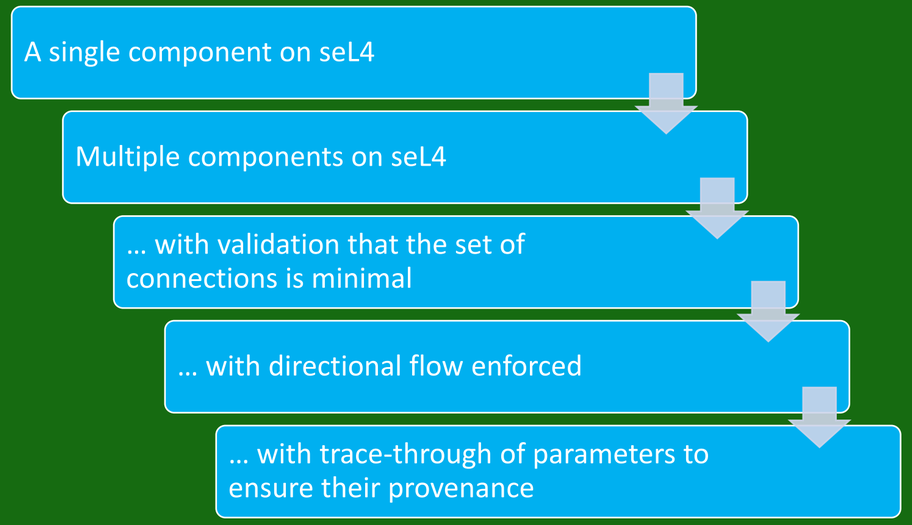
- Mapping the implementation to the design
- VisualCAmkES

- Flight planner retrieves GPS co-ordinates, then controls the motors as required.



Using unproven hardware / features?

- Development and verification is ongoing:
  - MCS Kernel Extension
  - AARCH64 port



1. Developer behaviour
2. Mapping the design to the implementation

Core Platform?

Improving the uptake of seL4

Pros and cons of developing on seL4

- seL4 can deliver significant benefit to many applications
- Formal verification = unique selling point
- It’s fast
- It’s open-source

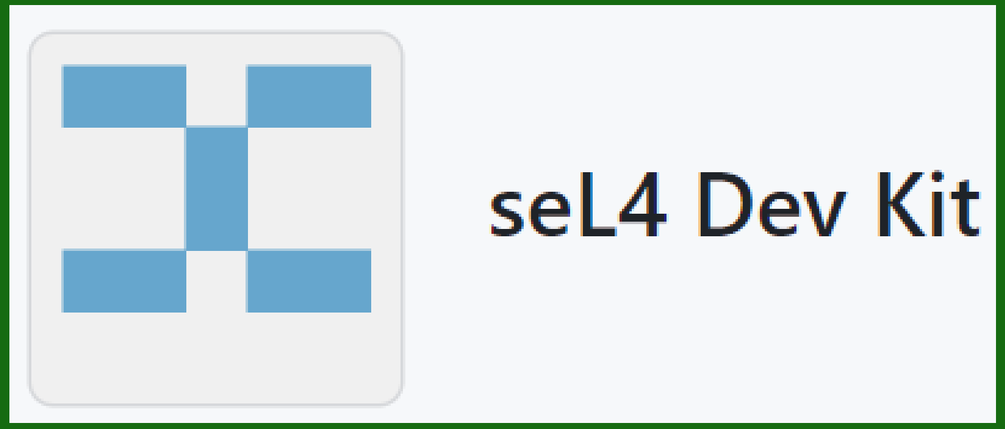
What are the problems?

- Development is complex
- Limited expertise
- Proofs may not be valid for your platform / features

seL4 Developer Kit

- Funded by the NCSC and developed by Capgemini Engineering

- Targeted at developers who…
  - Have some programming experience within a Linux, macOS, or Windows environment
  - Are familiar with seL4's concepts and benefits
  - Are looking for more practical and hands-on guidance to get started with seL4.



github.com/sel4devkit

Summary

Basic Requirements

Development Environment Setup

First Boot

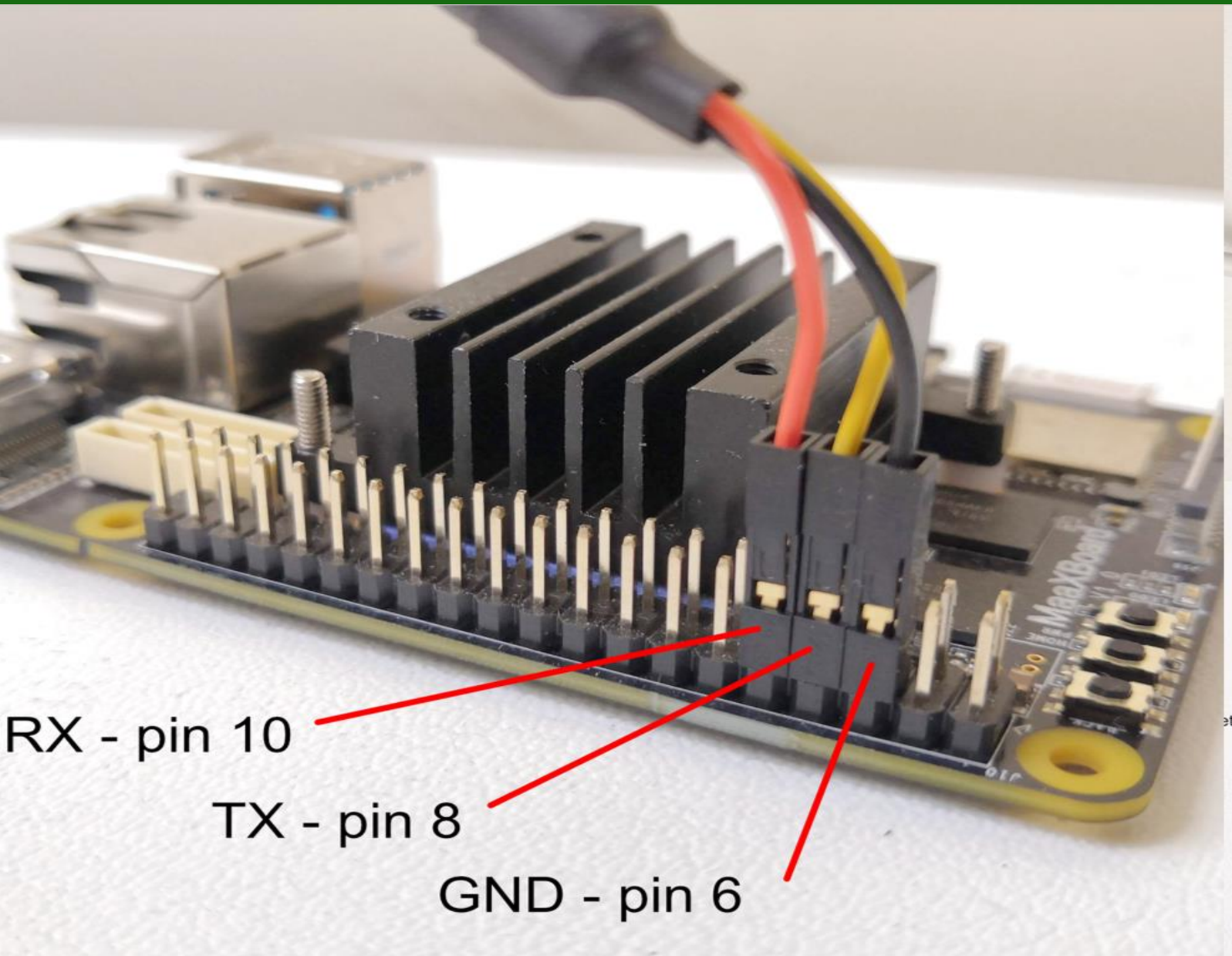
seL4 Application Development

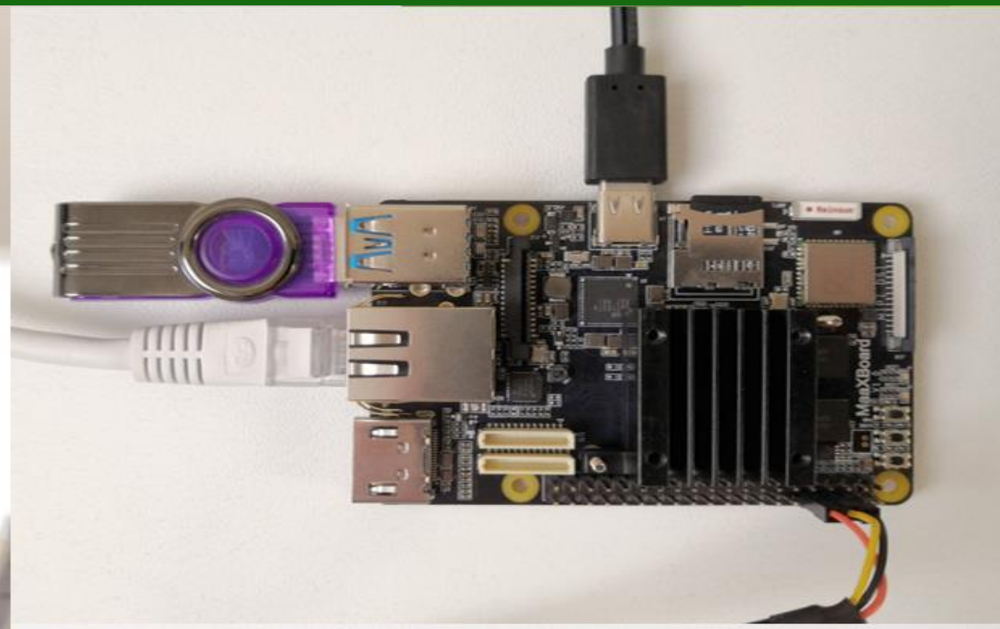
Device Driver Development

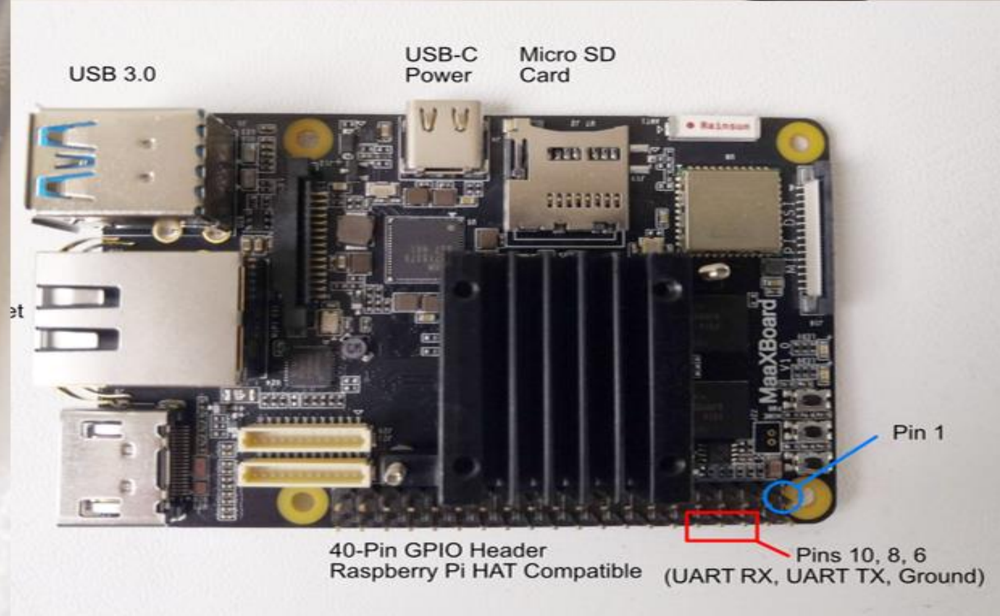
Case Study Application

Hardware Requirements

Item	Notes	Order Code
Avnet MaaXBoard (AES-MC-SBC-IMX8M-G)	Mandatory	3436577
USB-to-TTL Serial UART Cable	Mandatory	2147356
16GB Micro SD Card	Mandatory	3498607
USB Micro SD Card Reader/Writer	Mandatory	3493850
15W USB-C Power Adapter	Mandatory	3106255
USB Flash Drive	Optional - USB transfer only	General[1]
Ethernet Cable	Optional - TFTP transfer and some of the test applications	General[1]
USB Keyboard	Optional - some of the test applications	1848111[2]
SPI Bus Pressure Sensor	Optional - test application only	See SPI sensor appendix


RX - pin 10
TX - pin 8
GND - pin 6




USB 3.0
USB-C Power
Micro SD Card
Pin 1
40-Pin GPIO Header
Raspberry Pi HAT Compatible
Pins 10, 8, 6
(UART RX, UART TX, Ground)

Pros and cons of developing on seL4

- seL4 can deliver significant benefit to many applications
- Formal verification = unique selling point
- It’s fast
- It’s open source

What are the problems?

- Development is complex
- Limited expertise
- Proofs may not be valid for your platform / features

Summary

- seL4 helps to reduce the scope and impact of bugs and vulnerabilities
- seL4 helps us to prevent the unknown things that shouldn’t happen
- You can scale up or down the assurance activities of an seL4-based software implementations, depending on the product context
- seL4 aids reviewing of software updates
- There is value beyond the proofs
  - Developer behaviours
  - Design to implementation mapping
- seL4 dev kit
  - github.com/sel4devkit

National Cyber Security Centre

Thank you

Gage – The National Cyber Security Centre