System Information Flow Analysis

Ihor Kuz

seL4 summit 2023

PROTECTION

RESILIENCE

LIFECYCLE

- Protection: Security
  - Discover and remove vulnerabilities
  - Design to prevent threats and vulnerabilities
- Information Flow is a critical part of this

Security and Information Flow

Information Flow and seL4

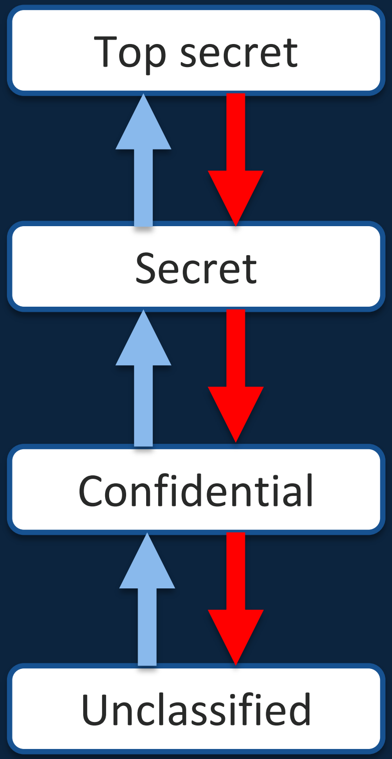
Static Systems

Dynamic Systems

Security and Information Flow

- Security Policy: what is allowed/disallowed
- Information Flow as part of security policy
  - Consider system as a communication graph
  - What information flows should be
    - allowed?
    - forbidden?
    - required?

Example: Clearance Levels

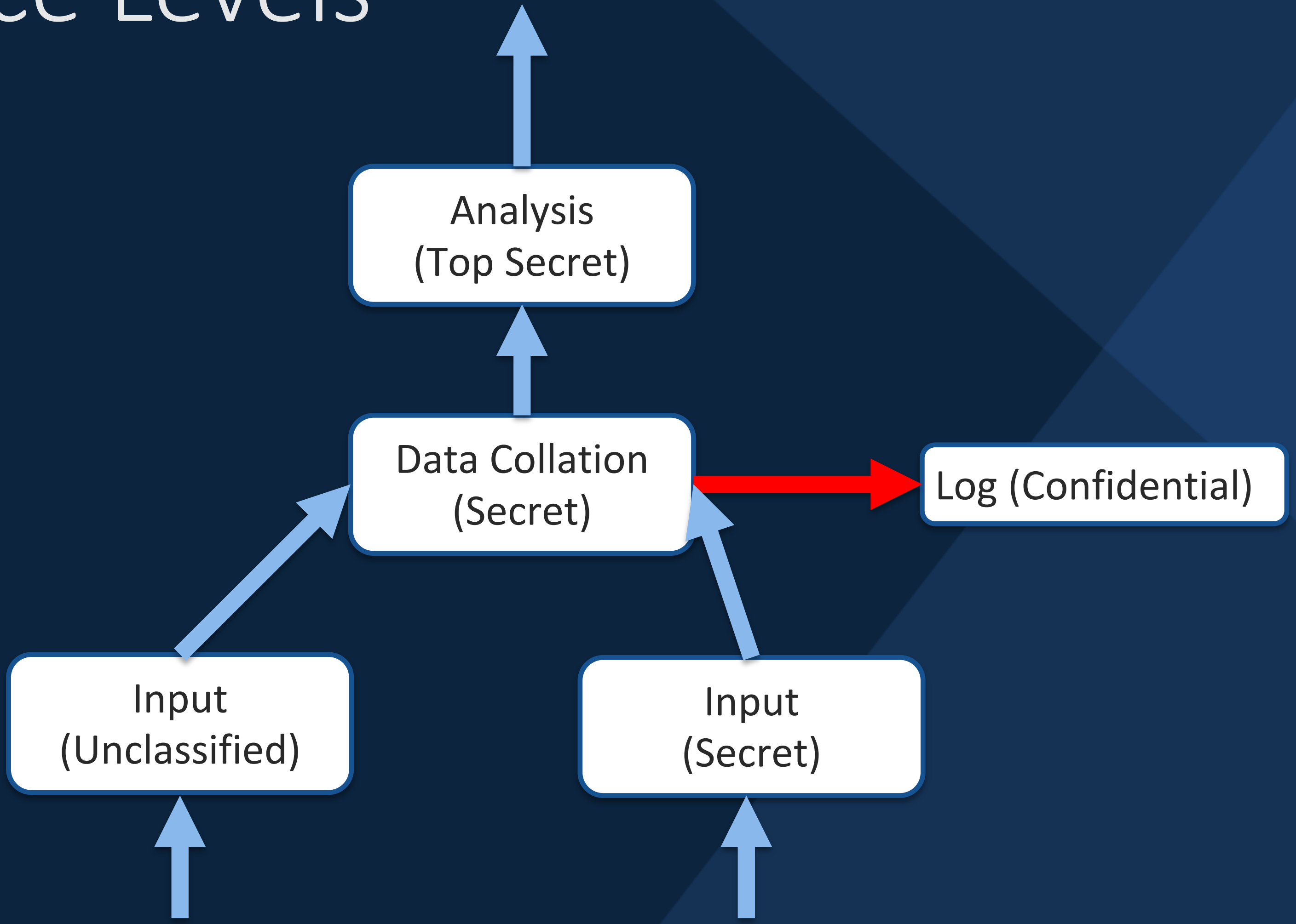


Top secret

Secret

Confidential

Unclassified



Analysis (Top Secret)

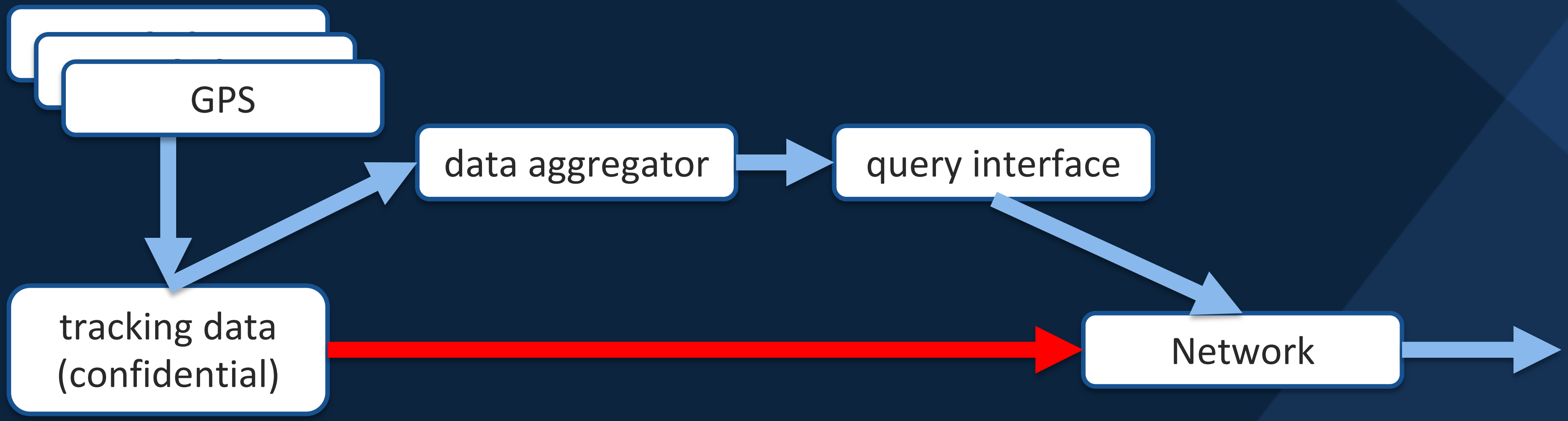
Data Collation (Secret)

Log (Confidential)

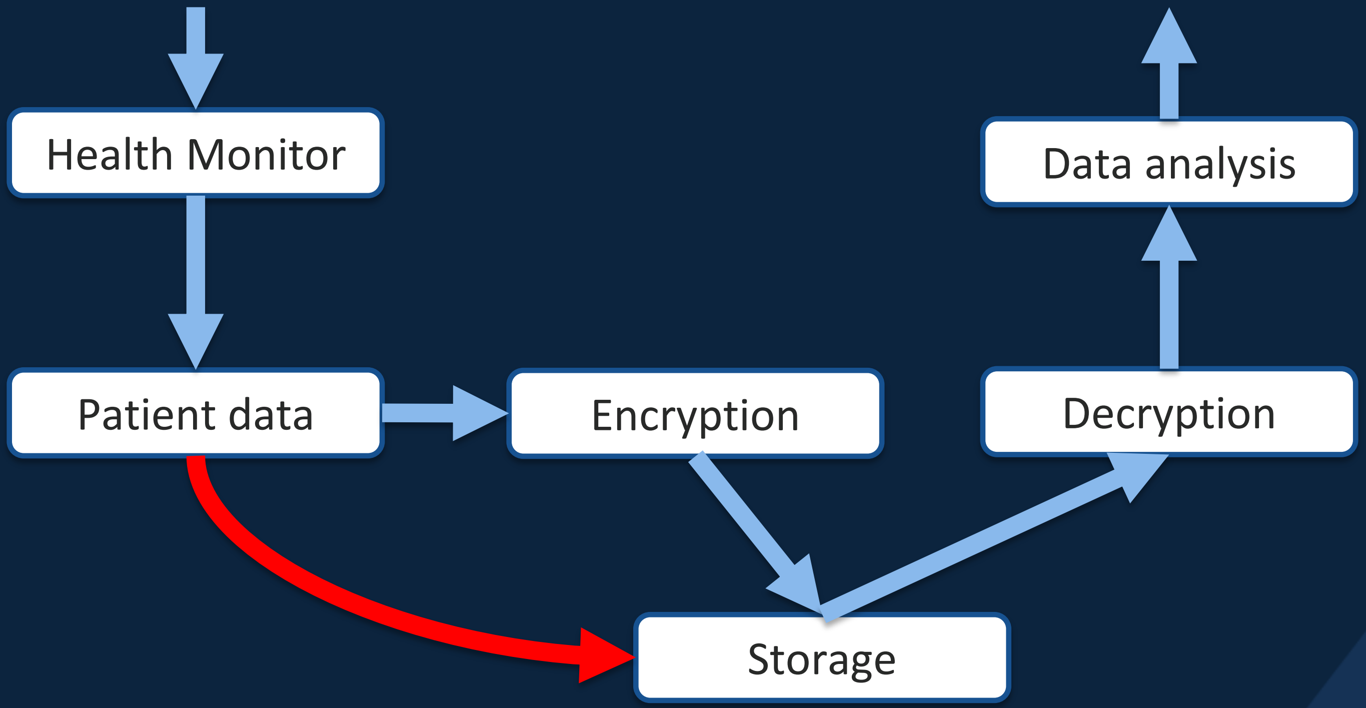
Input (Unclassified)

Input (Secret)

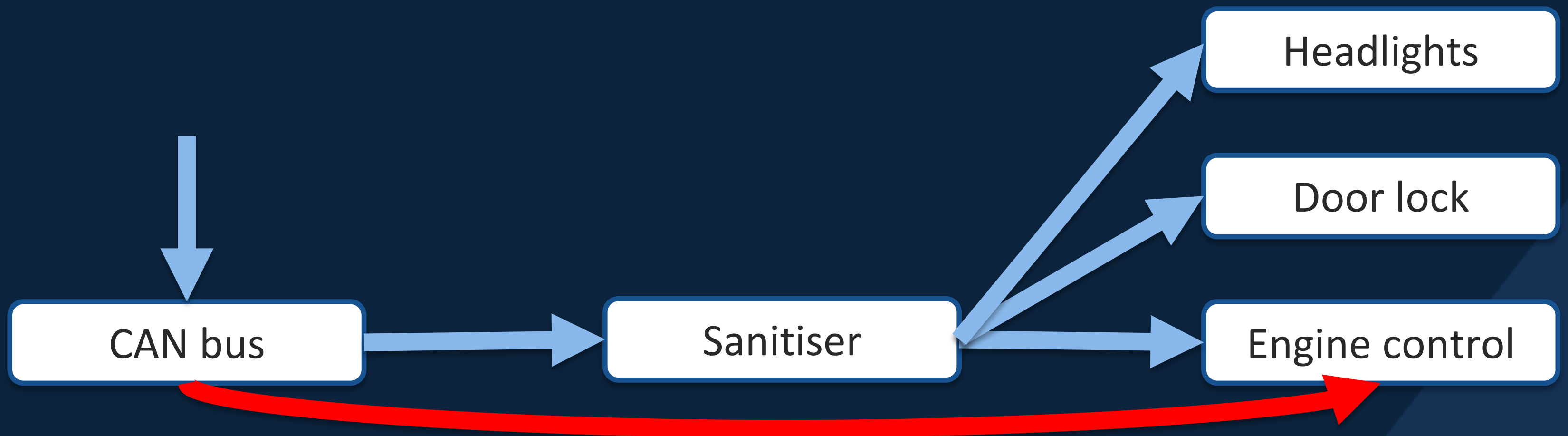
Example: Confidential Data



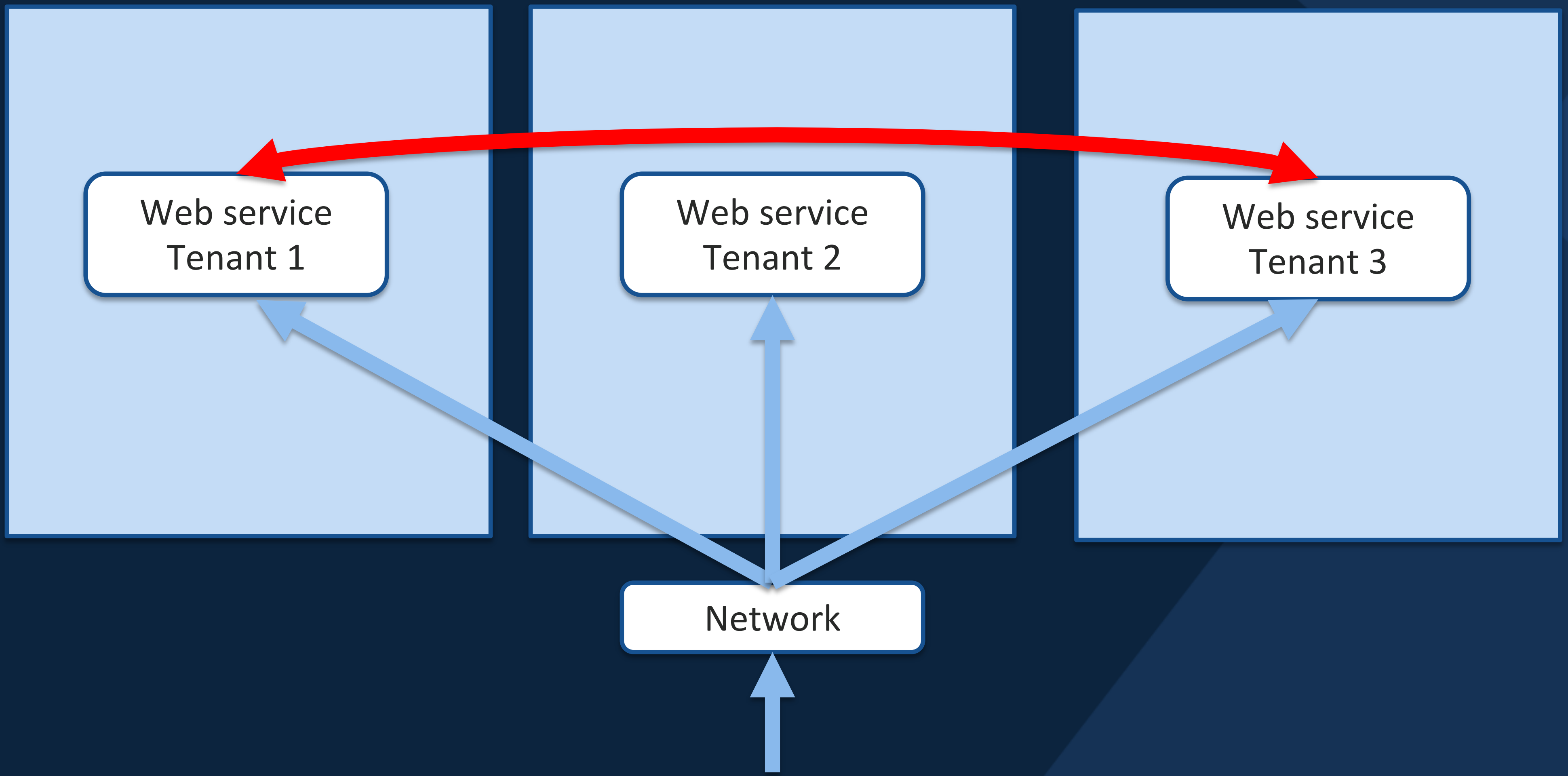
Example: Encrypted Data



Example: Sanitised Input



Example: Multi-tenancy



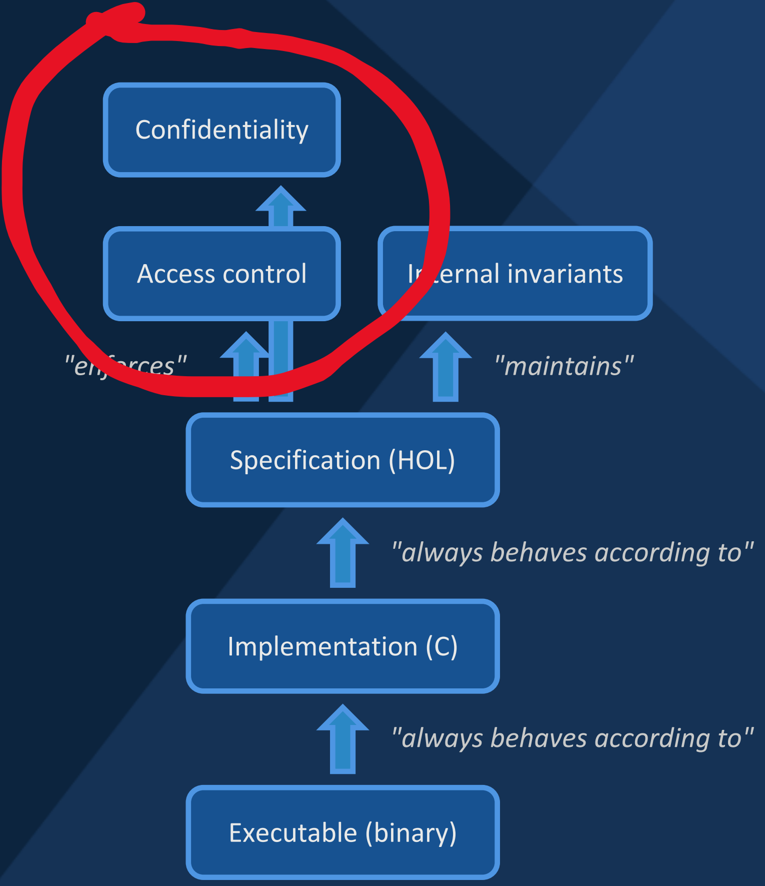
Formal Information Flow

- Security Model:
  - Formal model, property, policy, mechanism, analysis, assurance

- Multi-level Security (MLS): policy: no write down, no read up
  - Bell-LaPadula (C), Biba (I)
- MILS: separation kernel, mechanisms, no flow policy
- Separation:
  - Non-interference: traces - actions of high can’t affect output of low
  - GWV separation: allowed communication between memory segments
- Take-grant: access control model
- Data flow graph analysis

Information Flow and seL4

- seL4 security proofs
  - Integrity/Access Control: Take grant
  - Confidentiality: (intransitive) Non-interference
- Limitations
  - Domain scheduler
  - Cap transfer limitations
    - No cap transfer after initialization (C)
    - Grant => same label
    - Call requires Grant
  - No interrupts



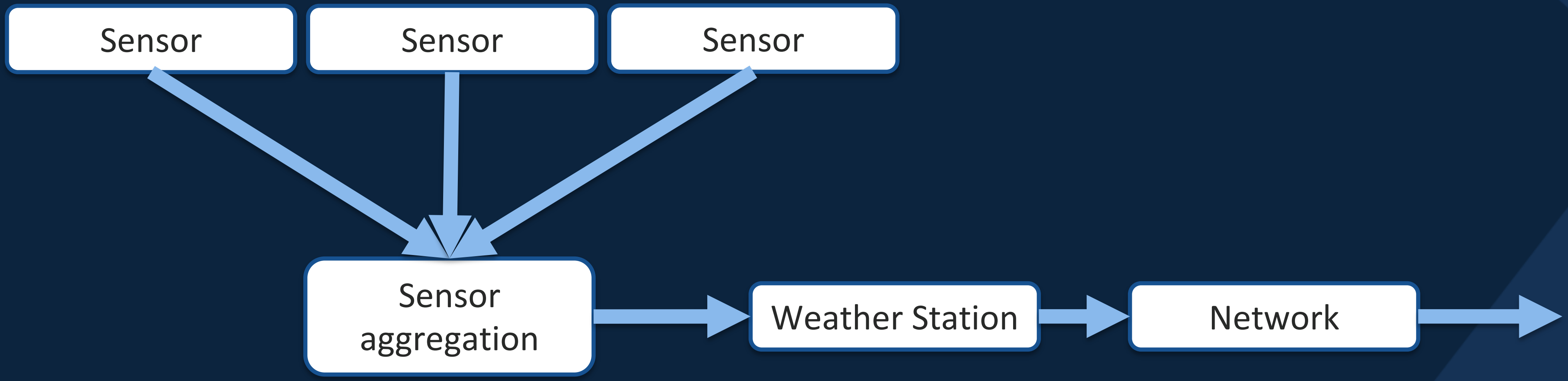
Information Flow in Static Systems

- Static System
  - Resources allocated at init
    - Processes, communication channels, memory access, etc.
  - Never change
- Analysis
  - Formal analysis (non-interference, etc.)
  - Graph analysis
- Rely on seL4 security properties
  - Communication only through defined channels

Information Flow in Dynamic Systems

- Static systems too limiting
- Dynamic Systems
  - Change over time
- Categories
  - Semi-static: static architecture - restart components, connections
  - Semi-dynamic: replace static architecture
  - Fully Dynamic: create/destroy components and connections

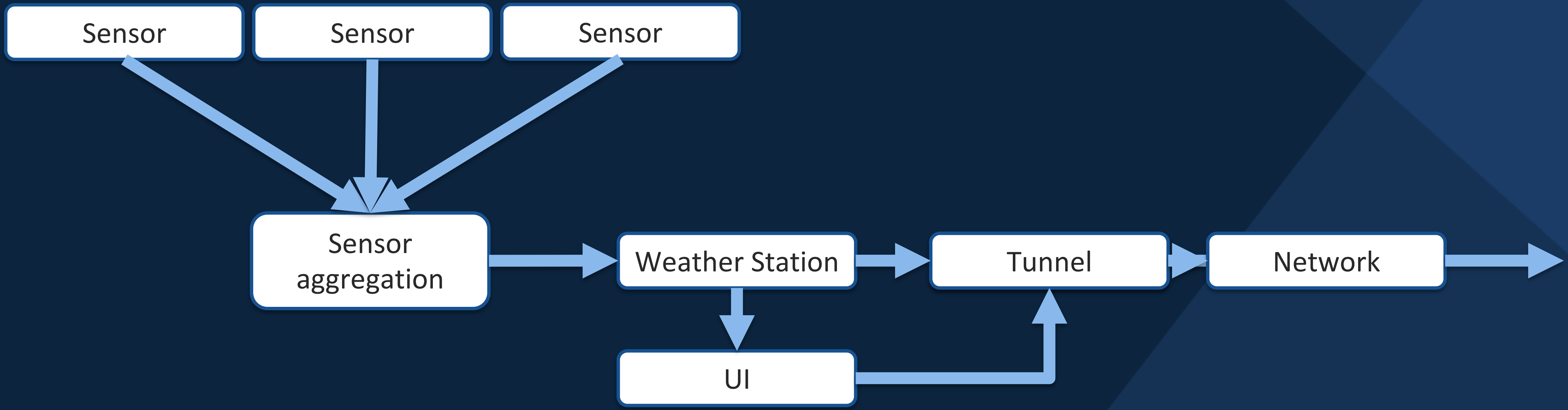
Semi-static Example



Semi Static Analysis

- Static graph analysis
- Component/Connection restart: must ensure
  - Flow doesn’t change: same before and after restart
  - Restarted component has same resources
  - Restarted connections connect same components
  - System state during restart doesn’t violate policy

Semi-dynamic Example

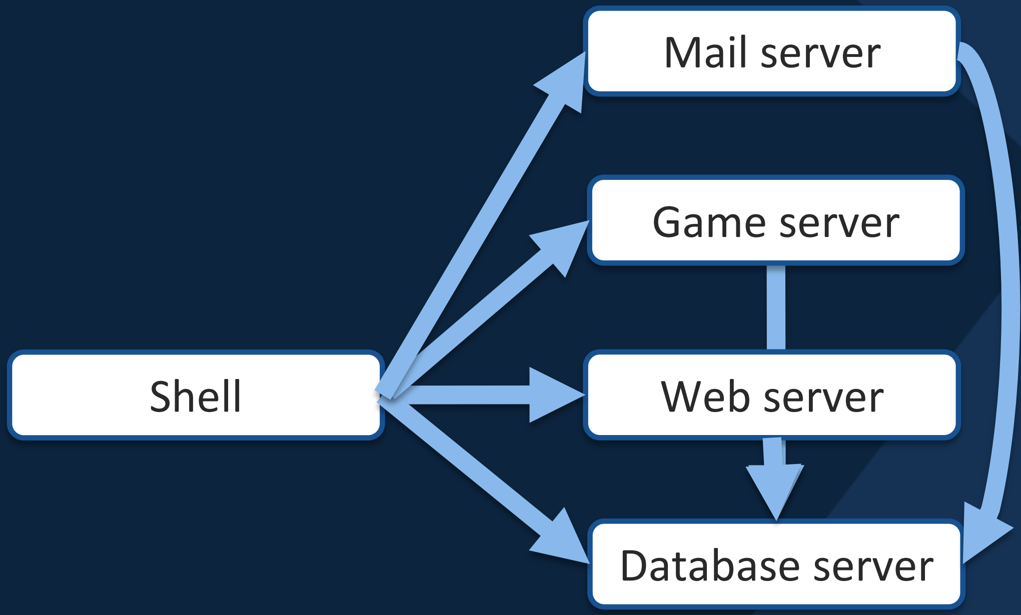


Semi-Dynamic Analysis

- Switching static architecture
  - Ensure new architecture doesn’t violate policy
  - Admission check:
    - architecture analysis: online or offline
  - Ensure system between architectures doesn’t violate policy
    - Like system init for static systems
- Challenges
  - Dealing with state while switching
  - Partial switch (only switch components that change)

Fully Dynamic Example

- ➢ start webserver.elf
- ➢ start database.elf
- ➢ connect webserver database
- ➢ start mailserver.elf
- ➢ connect mailserver database
- ➢ stop webserver.elf
- ➢ start gameserver.elf
- ➢ connect gameserver database

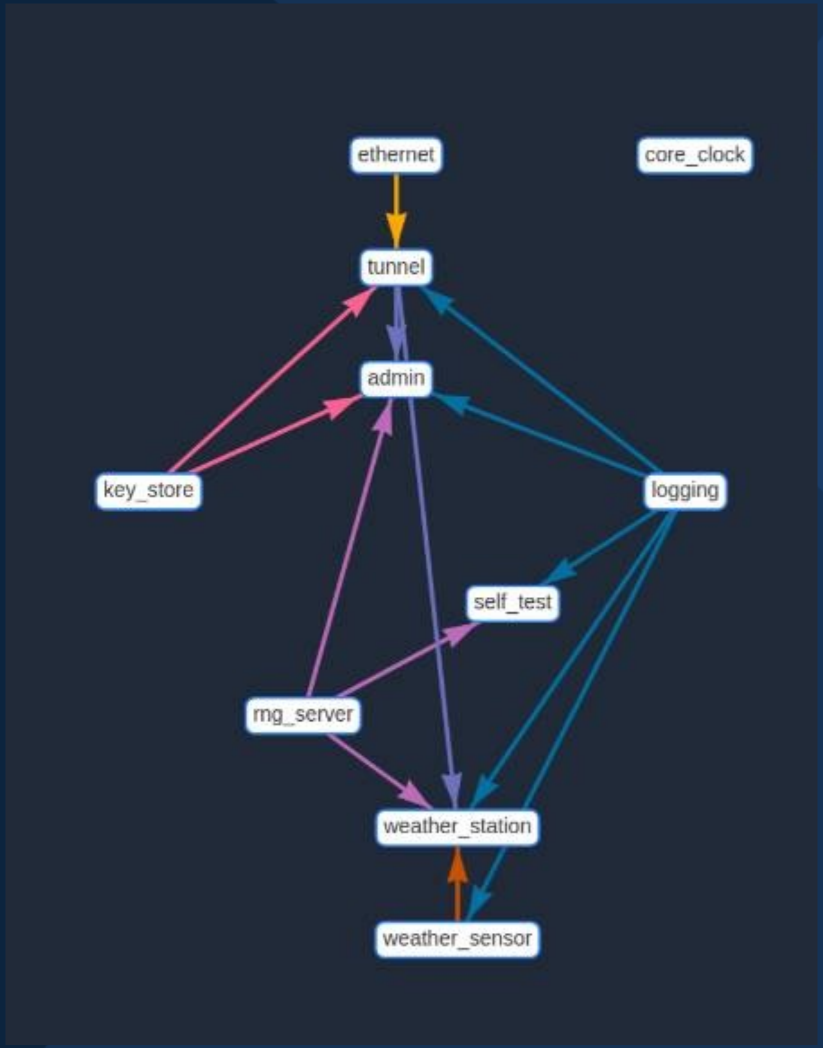


Fully Dynamic Analysis

- Single controller vs Decentralised control
  - Can ***any*** component create new components?
- Challenges
  - Monitor when components create new components/connections
  - Maintain internal model of graph
  - Analyse graph at run-time
  - Verify all components that create new components/connections
    - Ensure they don’t violate information flow policy

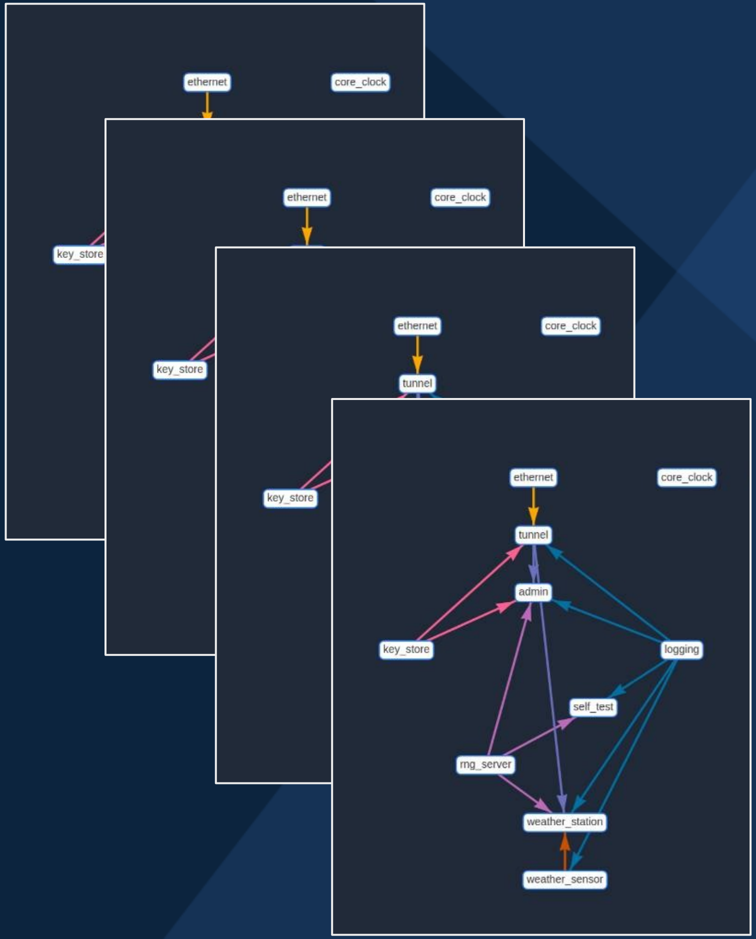
Kry10 OS

- Semi-dynamic system
- Manifest. defines:
  - components, allowable connections
- Dynamic features
  - Component restart
  - Component update
  - Dynamic connections
  - System update



Information Flow Analysis for Kry10 OS

- Static Graph Analysis
  - Based on manifest description
- System as a succession of Static Graphs
  - Admittance checks
- Key challenges
  - Root component
    - Loading: system == manifest
    - Restarting: doesn’t change graph
  - Message server
    - Connections are as expected
  - What do we need from seL4?



Conclusion

- Information Flow is important part of Security
- Many formal models, often too strong
- Static Systems
  - Information flow analysis as graph analysis
- Dynamic Systems
  - Reuse static graph analysis when possible
  - Kry10 working on semi-dynamic systems with graph analysis