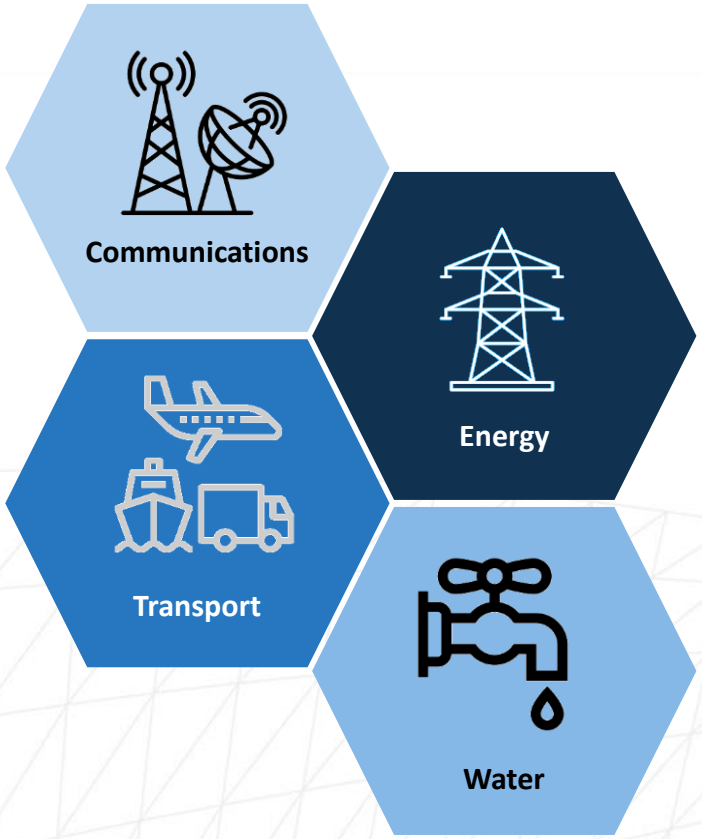
Characterising Software Failure and the Consequences of Software Failure for Critical National Infrastructure Resilience

2025

CNI AND DIGITALISATION



(+ other 9; 13 CNI sectors in the UK in total)

- CNI are *"open complex interdependent systems"* (UNDRR, 2022)
- Digitalisation
  - ✓ Remote access and better maintenance;
    Reduced costs and increased flexibility for operators
  - ✗ Increased vulnerabilities (open door to threats from software failure)
- Roadmap to 2030 (IPA, 2021):
  - Use modern digital approaches and technologies
  - Digital-by-default infrastructure delivery → Collaborative approach using digital technologies to improve productivity, efficiency and quality

TRIGGERS THAT LEAD TO SOFTWARE FAILURE

		Categories of software failure:
Attacks	→	1. Security and resilience failures in software
Software defects	→	2. Data driven
Data handling	→	3. Software-Hardware interface driven
Interface issues	→	4. Operational (HCI)
Human Computer Interface (HCI) issues	→	5. Intrinsic

TRIGGERS THAT LEAD TO SOFTWARE FAILURE

Category	Sub-categories
Security and Resilience Failures	1.1. Adversarial attacks
	1.2. Ransomware attacks
	1.3. Environmental hazards
Intrinsic Software Failures	2.1. Demand defect / software requirements are incorrect
	2.2. Functional and performance defect
	2.3. Software structure defect
	2.4. Software implementation and coding defect
Data-Driven Failures	3.1. Data handling issues
	3.2. Data defect / quality
Software-Hardware Interface Failures	4.1. Internal and external interfaces are incorrect
	4.2. Lack of coordination between the interface
	4.3. I/O timing error (causing mis-match and duplicates)
Human-Computer Interface Failures	5.1. User interface defect
	5.2. Operator error

CONSEQUENCES AND IMPACT TO CNI

"a random failure […] in a component of an interdependent system could cause cascading effects that can potentially collapse a component of or the entire system of interdependent CI"

(Palleti et al., 2021)

- Malfunctions / Degraded performance
- Operational disruption
- Lost of public trust
- System crashes and unexpected downtime
- Equipment damages
- Economic impact / Increased maintenance costs
- Safety hazards

CASE STUDIES
1 – NATS Air traffic control failure

(BBC, 2023)

Date: 28th August 2023

Issue: Inability of the system software to successfully process the flight plan data for a specific flight

Trigger: Waypoints identifiers confusion

DVL: 1) Devil's Lake, North Dakota, US;
2) Deauville, France

Exception handling failure → Intrinsic software failure

Degree of impact:

Over 700,000 passengers (+ others affected) → Considerable financial and emotional consequences for them

CASE STUDIES
2 – Attack on Colonial Pipeline

Date: 6th – 12th May 2021

Issue: Victim of a ransomware attack

Security and resilience

- Possible root cause: An exposed password for a VPN account allowed access from cyber-attackers
- Ransomware infected the IT network → multiple computer systems were affected

Degree of impact:

- Airline industry was affected due to jet fuel shortage
- Panic-buying from citizens
- Spike in average price
- Economic losses affecting various sectors reliant on fuel for transportation and operations
- Ransom payment ($4.4 million in cryptocurrency)

OTHER CASE STUDIES

UK Royal Mail (2023)	• Ransomware attack → Disrupted international mail deliveries • Vulnerabilities in logistics infrastructure → Urgency of software resilience in postal and supply chain industries
UK Railway Cyberattack (2024)	• Cyber-vandalism → Compromised Wi-Fi networks at 19 UK railway stations

DISCUSSIONS WITH STAKEHOLDERS – KEY FINDINGS

Software vs. Service Resilience	Architecture & design vs. minimization of lost user hours & financial stability
Interconnected Systems	Digitalisation increases complexity & risk of cascading failures
Failure Impact	Small software defects can trigger major disruptions
User-Centric Approach	Stronger operator-user relationships improve resilience
Legacy Risks	Outdated software poses security threats & requires better governance
Cybersecurity and Engineering	Poor validation & design flaws weaken security
Data issues	Useful digital tools for failure detection (e.g., Digital Twins) but unreliable with incomplete data
Insurance and Regulation	Software lacks structured risk management; policy evolution needed

STRATEGIES FOR ENHANCING SOFTWARE RESILIENCE IN CNI

Secure-by-Design

Data quality management

Improve operator training and usability testing

Stakeholder engagement

Metadata and ontologies for model transparency

Integration and interoperability

Stronger insurance and regulatory frameworks

Adopt Software Bill of Materials (SBOM) tracking

CONCLUSION

Building a Resilient Digital Future

- **Software resilience is critical** for CNI as digitalisation accelerates
- **Key strategies:**
  - ✓ Secure design
  - ✓ Strong data management and collaboration.
- **Cybersecurity must evolve** → High-profile failures highlight urgent action needed
- **Future focus:**
  - ✓ Advanced security frameworks
  - ✓ Rapid incident response
  - ✓ Cross-sector cooperation.

Key Takeaways

1) **Detect failures early in highly-interdependent systems**
2) **Software Bill of Materials (SBOM) to identify vulnerabilities and support risk management**
3) **Regulations & insurance must adapt to software resilience needs**