Challenges in Edge Computing II: Security/Privacy, Deployment, Foundations

Chapter 5

Topic: Security | Service Management | Deployment | Foundations

Outline

- ❑ Security & Privacy
  - Sensor security
  - ShadowNet, TEEs, KLRA
  - Blockchain data sharing
- ❑ Service Management
  - The DEIR model
  - Differentiation, extensibility
  - Isolation, reliability, EdgeWare
- ❑ Deployment
  - Edge node placement
  - LF Edge, EdgeX, Akraino
  - HW / SW selection
- ❑ Foundations & Business
  - CAP theorem
  - Multi-objective optimization
  - Business models & challenges
- ❑ Summary & Practice

Security & Privacy

Three Privacy Challenges

- Homes and IoT systems capture enormous private data. Computing near the source protects it but three challenges remain.

Awareness in society

Users under-protect data: 49% of home WiFi is insecure, 80% keep default router passwords, 89% of public hotspots are insecure.

Data collector = owner

Devices both collect and own data. Keep it local, let users control provider access, and delete highly sensitive data.

Lack of tools

Edge devices are resource-limited and highly dynamic; existing security methods don’t fit. Platforms like Open mHealth are early steps.

Weisong Shi et al. “Edge computing: Vision and challenges”. In: IEEE Internet of Things Journal 3. 5 (2016), pp. 637–646.

Security & Privacy: Sensor Security

Attacking the senses of the edge

- Autonomous vehicles and smart devices perceive the world through sensors — attackers manipulate or obstruct sensor data without ever touching the compute system.

Cameras

Fooled by counterfeit signs, signals or fake objects; blinded by high-brightness IR lasers.

GNSS / INS

GNSS jammed or spoofed by fake signals; INS misled by strong magnetic interference.

LiDAR

Deceived by absorbent/reflective surfaces or manipulated laser pulses → wrong distances.

Ultrasonic / Radar

Passive perception & last-line defense — spoofed & jammed by specialized signal generators.

Security & Privacy: Network & Platforms

Defending edge networks & systems

ShadowNet

Bhardwaj et al. — edge functions monitor IoT traffic to preempt IoT-DDoS. Detects 10× faster, blocks 82% of traffic before it hits the Internet.

SDN security

Yi et al. — SDN improves monitoring, intrusion detection and access control across the edge network.

TEEs

Ning et al. — Intel SGX, ARM TrustZone & AMD SEV bring trusted execution to heterogeneous edges with minimal overhead.

KLRA

Li et al. — Kernel-Level Resource Auditing for IoT/edge OSs; watches system behavior and raises alerts on anomalies.

Ketan Bhardwaj, Joaquin Chung Miranda, and Ada Gavrilovska. "Towards IoT-DDoS prevention using edge computing". In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). 2018.

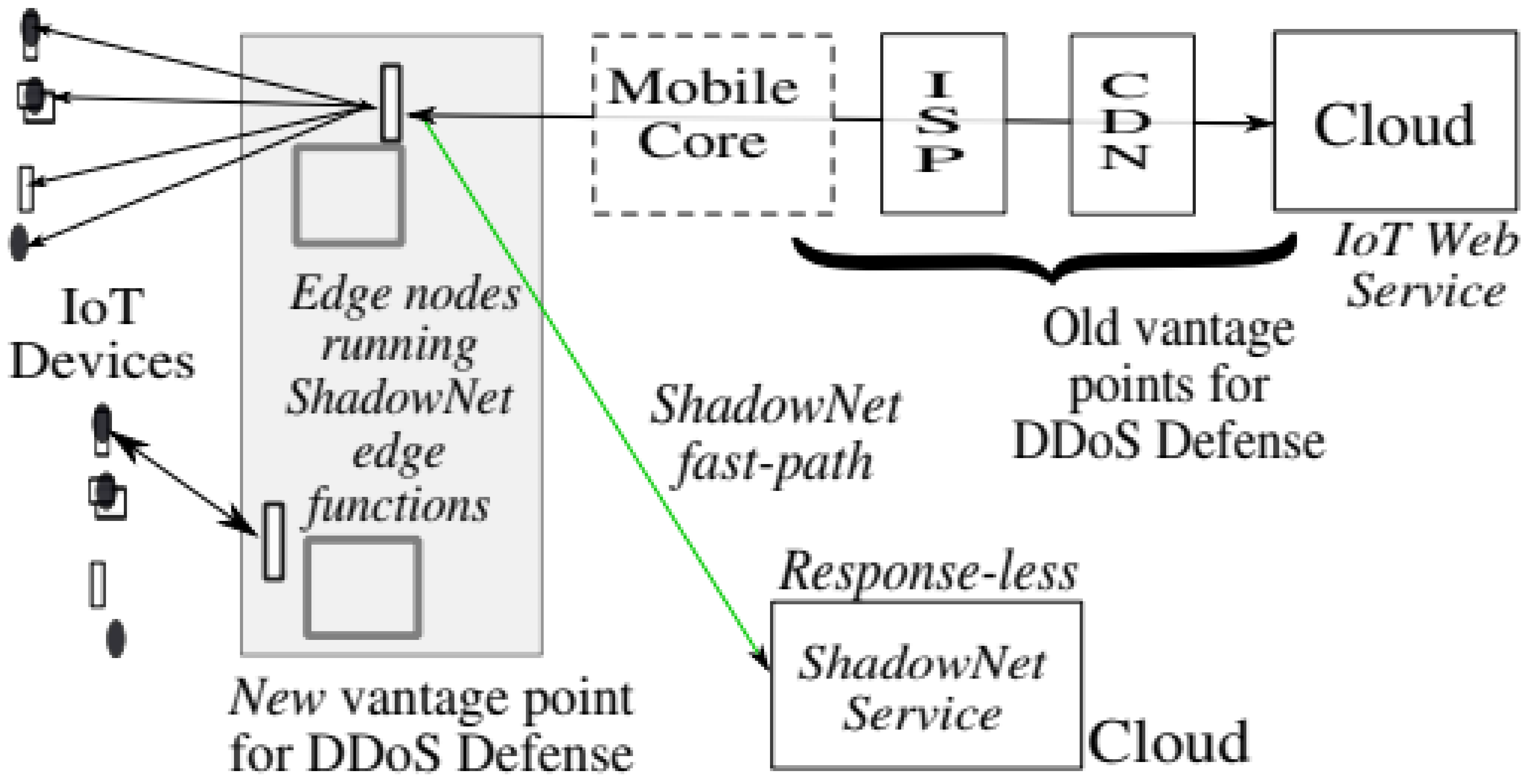
Shanhe Yi, Zhengrui Qin, and Qun Li. "Security and privacy issues of fog computing: A survey". In: Wireless Algorithms, Systems, and Applications (WASA). Springer. 2015, pp. 685–695.

Zhenyu Ning et al. "Preliminary study of trusted execution environments on heterogeneous edge platforms". In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. 2018, pp. 421–426.

Peng Li et al. "KLRA: A kernel level resource auditing tool for IoT operating system security". In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. 2018, pp. 427–432.

Security & Privacy: ShadowNet

Defending edge networks & systems



ShadowNet Overview Diagram ([Source](#))

➢ Key objectives

Edge-based DDoS detection moves attack detection closer to IoT traffic sources instead of relying only on centralized cloud defenses.

Application-level IoT DDoS attacks, where malicious traffic can appear legitimate and is difficult for traditional defenses to identify.

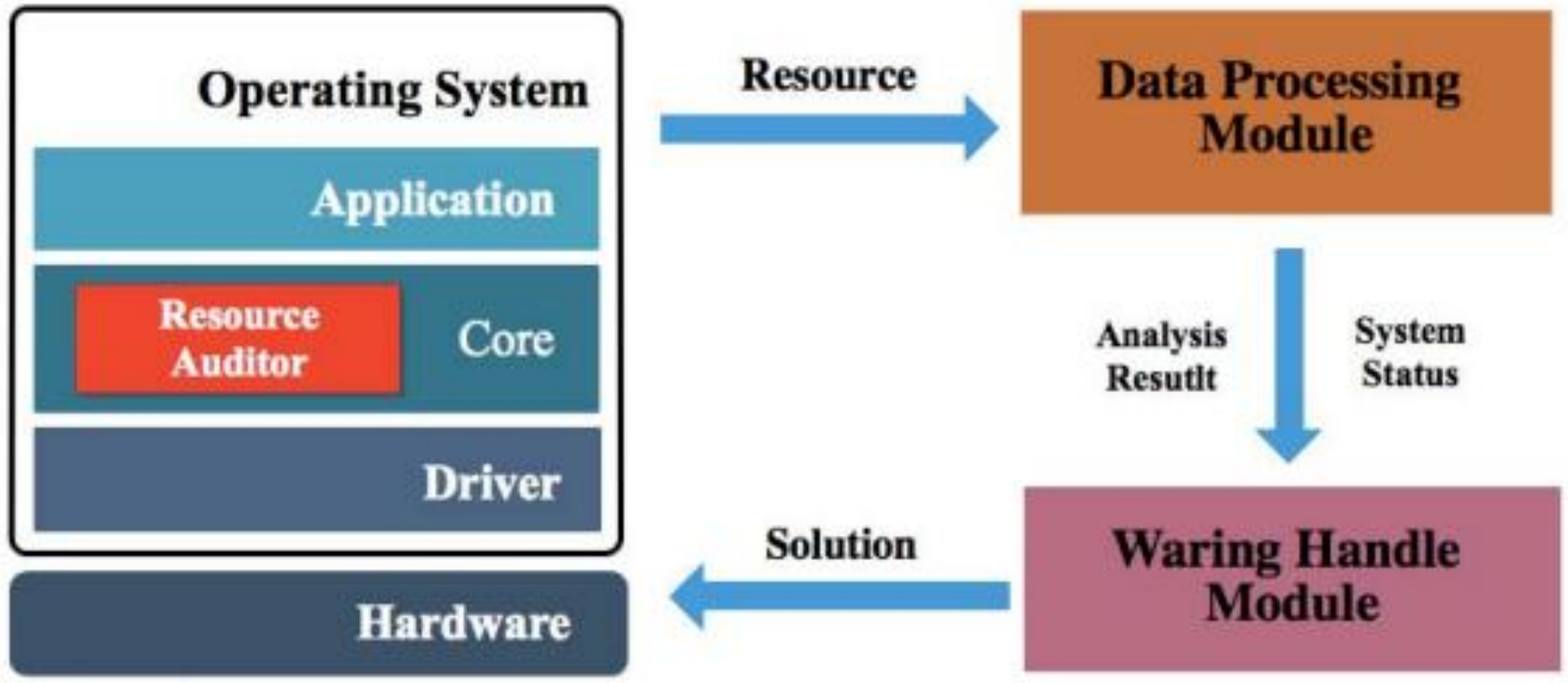
Distributed edge infrastructure to detect and stop attacks earlier, reducing their impact before traffic reaches core Internet services.

Emphasizes faster detection and traffic reduction, reporting up to 10× faster detection in preliminary results.

Ketan Bhardwaj, Joaquin Chung Miranda, and Ada Gavrilovska. "Towards IoT-DDoS prevention using edge computing". In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). 2018.

Security & Privacy: KLRA

Defending edge networks & systems



KLRA Overview Diagram ([Source](#))

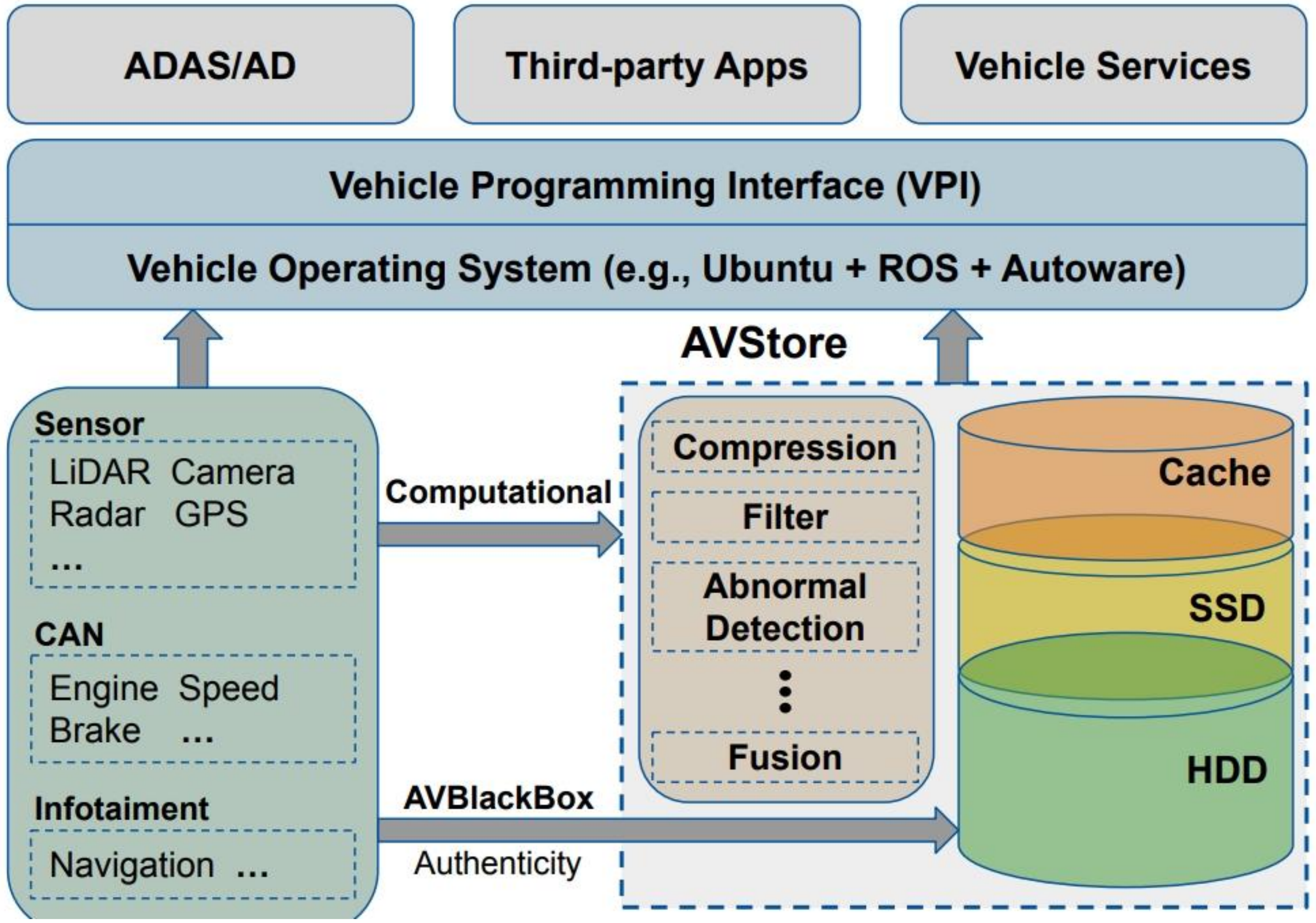
➢ Key objectives

- **Kernel-level auditing** monitors security-sensitive resource events directly inside the IoT operating system kernel.
- **Resource consumption monitoring** tracks how system resources are used and identifies abnormal usage patterns.
- **Behavior-based detection** analyzes resource-usage patterns that may indicate malicious or abnormal activities.
- **Edge/IoT security** focus improves device-level visibility before threats propagate through the larger edge infrastructure.

Peng Li et al. "KLRA: A kernel level resource auditing tool for IoT operating system security". In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. 2018, pp. 427–432.

Security & Privacy: Data Sharing

Blockchain for Secure Sharing



Storage system architecture for autonomous-vehicle data

Edge integration breaks data silos but complicates secure sharing — many interfaces widen the attack surface, and weak devices can't run heavy security algorithms.

➢ Blockchain in edge computing

Tamper-proof storage
Blockchain makes edge data tamper-proof; IPFS stores data off-chain, keeping only the hash on-chain to shrink volume.

New business models
Enables monetizing edge services — but systematic integration research is still lacking.

Access control
Combine attribute-based access control with blockchain — but public visibility to all nodes raises privacy concerns.

Yuxin Wang et al. "Quantitative analysis of storage requirement for autonomous vehicles". In: Proceedings of the 16th ACM Workshop on Hot Topics in Storage and File Systems. 2024, pp. 71–78.

Service Management: The DEIR Model

Any reliable edge system exhibits four characteristics — Differentiation, Extensibility, Isolation, Reliability.

D

Differentiation

Services have different priorities — device diagnostics & failure alarms, or fall/heart-failure monitoring, outrank entertainment.

E

Extensibility

IoT devices are highly dynamic — new or replacement devices must integrate seamlessly into existing services.

I

Isolation

One app's failure shouldn't crash shared resources; removing an app shouldn't break the device; third parties can't reach private data.

R

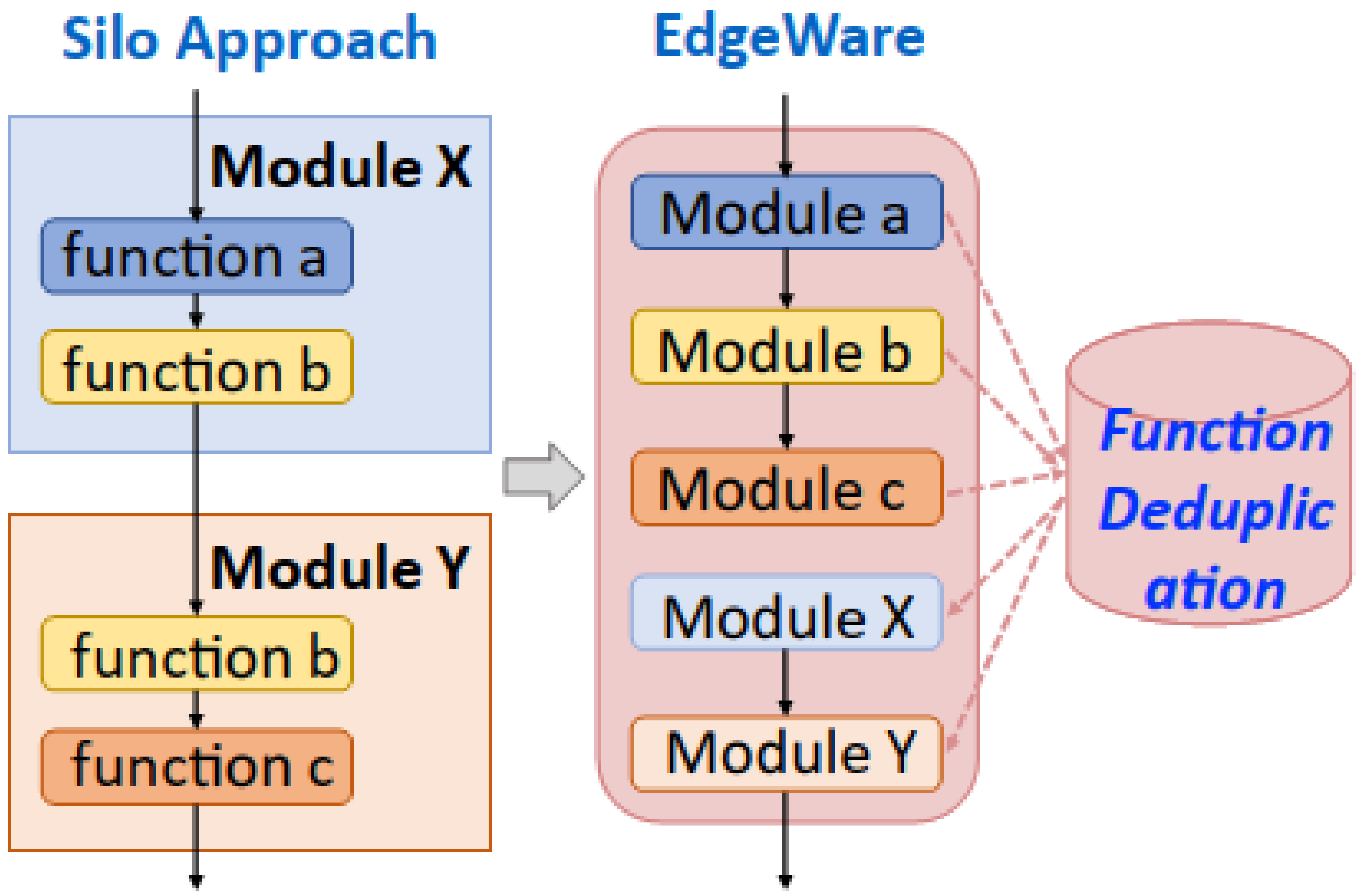
Reliability

From service, system and data views — detect failures, keep topology, and cope with unreliable sensing & communication.

Weisong Shi et al. "Edge computing: Vision and challenges". In: IEEE Internet of Things Journal 3. 5 (2016), pp. 637–646.

Service Management: Middleware

EdgeWare: managing edge services



Function consolidation & deduplication: services become reusable function modules

Function cache. A "function cache" manages the lifecycle of edge services, so limited resources dynamically support what's needed.

➢ EdgeWare: Four key features

Open-source, extensible, flexible middleware for edge service execution.

On-demand model switching Easily transition & upgrade ML models.

Consolidation & dedup Eliminate redundant recurring functions; maximize reuse.

Event-driven apps Reduce workload with event-triggered execution.

Dynamic workflows Extend functionality via customizable workflows.

Sidi Lu et al. "EdgeWare: Toward extensible and flexible middleware for connected vehicle services". In: CCF Transactions on High Performance Computing 4. 3 (2022), pp. 339–356.

Deployment & Integration: Node Deployment

Placing & trusting edge nodes

Selecting edge nodes

Choose nodes along the cloud-to-endpoint path to cut latency & bandwidth. Base stations can serve as nodes; direct backbone connections reduce delay.

Choosing edge data

Overlapping data sources give many solutions — e.g. vehicle speed from cameras, traffic lights or roadside units. Pick the source that minimizes latency & maximizes availability.

Ensuring reliability

Exposed nodes (e.g. smart cameras) fail in wind, snow or wear. Physical infrastructure & spatial-temporal backups protect device and data reliability.

Deployment & Integration: AI on the Edge

Deploying AI models under constraints

- Edge devices have limited compute & storage. Two complementary approaches make AI deployable — boost the hardware or shrink & split the model.

Intelligence devices

- General modules: CPUs, GPUs, FPGAs
- Customized AI processors per application
- More source data → higher accuracy & timely decisions
- Optimizing full edge-AI complementarity is still hard

Intelligence models

Model partition

Split the network across devices — independent, device-device, or device-server collaboration.

Model size reduction

Weight pruning removes low-impact neurons; quantization uses fewer bits to speed operations.

Deployment & Integration: Integration

LF Edge: Open-Source Deployment

- Edge devices have limited compute & storage. Two complementary approaches make AI deployable — boost the hardware or shrink & split the model.

EdgeX Foundry

IoT interoperability framework

- Modular reference architecture: ingest, normalize, analyze, share
- Protocols: MQTT, REST, BLE
- Enhanced security; smart cities, retail, manufacturing

Akraino Edge Stack

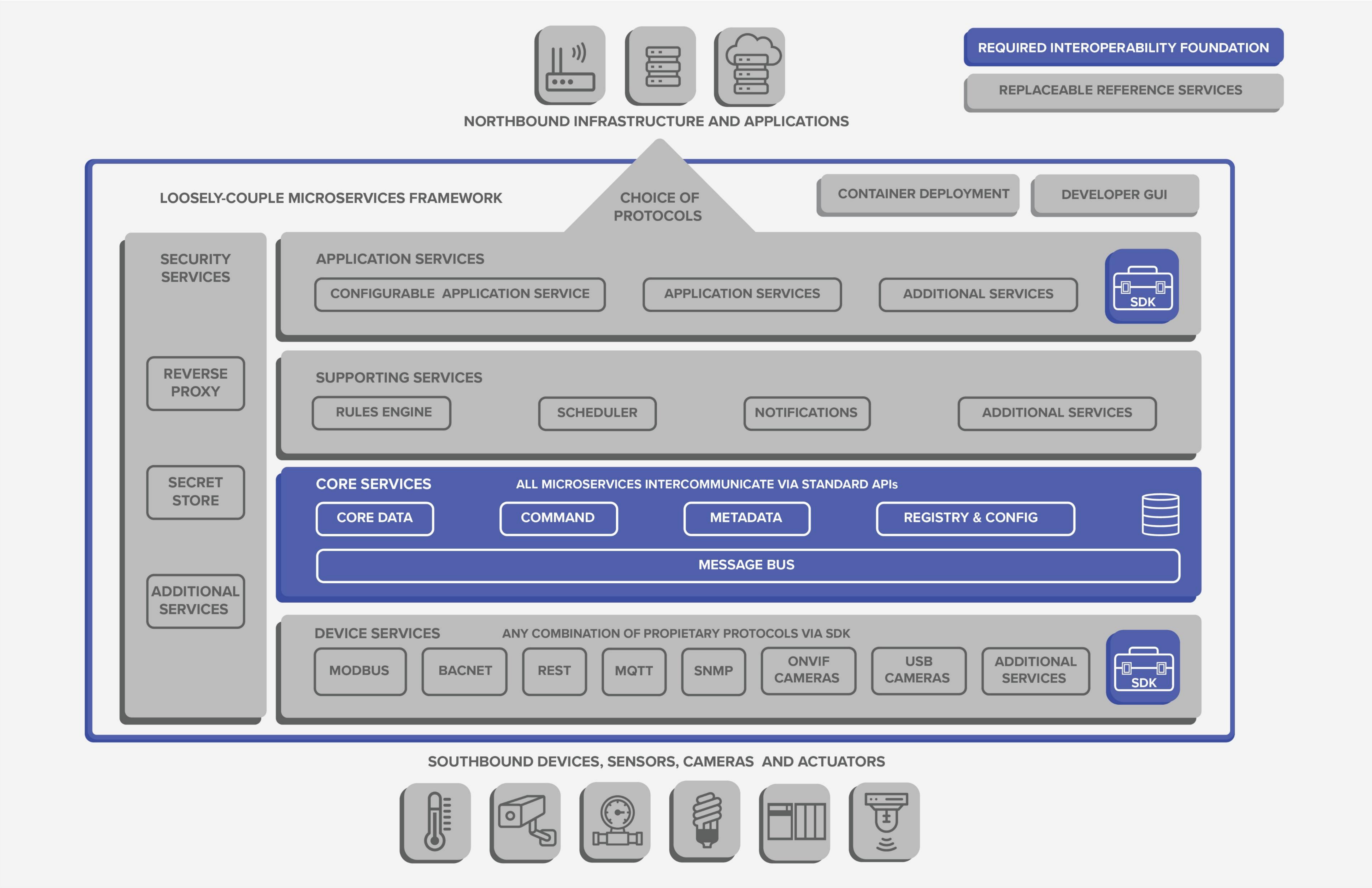
Blueprints for edge infrastructure

- Open-source blueprints for 5G, AI, IoT
- Low latency, high availability, scalability
- VMs, containers, microservices; zero-touch provisioning

The Linux Foundation. LF Edge: Building an Open Source Framework for the Edge. https://lfe dge.org/. Accessed: 2024-08-30. 2024.

Deployment & Integration: Integration

EdgeX Foundry



EdgeX Foundry platform architecture ([Source](#))

The Linux Foundation. Edge Foundary: An Open Edge Platform. https://www.edgexfoundry.org/. Accessed: 2026-08-29. 2026.

Deployment & Integration: Selection

Choosing Hardware & Software

- Edge systems are fragmented and heterogeneous. Finding the optimal model–framework–hardware combination means real cost and trial-and-error.

Hardware variety

CPUs, GPUs, FPGAs, ASICs — and huge spread even within a family (NVIDIA Drive PX2 vs. Jetson TX2).

Software frameworks

TensorFlow, PyTorch and others — each performs differently on different hardware.

Combinatorial choice

Many models × frameworks × hardware, each with distinct accuracy, power and cost profiles.

Need for tooling

Analyze compute needs first; tools that profile performance & power and recommend combos are badly needed.

Foundations & Business: Theory

Theoretical Foundations

CAP theorem

Brewer's hypothesis (PODC 2000), later proven — a cornerstone of distributed systems, especially distributed storage.

Consistency Every read sees the latest write.

Availability Every request gets a response.

Partition tolerance Works despite network splits.

Multi-objective optimization

A mature theory is still emerging. It must jointly optimize across four dimensions:

Computing

Load-balancing theory across cloud & edge.

Communication

Maximize edge-cloud transmission bandwidth.

Storage

Distributed storage foundations.

Energy

Lyapunov-based efficiency & reliability models.

Foundations & Business: Business

From Service-Driven to Data-Driven

Cloud model

Service-driven & unidirectional

Users buy on-demand, scalable, virtualized resources — IT infrastructure, software, services — center-to-user.

Edge model

Data-driven & multilateral

Spans IT + CT and many roles (platforms, connectivity, chips, sensors, apps). Requests submit data to owners — user-to-center and user-to-user.

The open challenge.

Edge value depends on many stakeholders. The Firework model reframes each request as a data request to its owner, shifting from a one-way center-to-user flow to a multilateral one. Integrating existing cloud business models into a workable multilateral edge model is the key unsolved problem.

Technical Challenges in Edge Computing

Challenge	Description
Hardware and software selection	Selecting appropriate hardware and software considering computational requirements, performance, and power consumption.
Integration with vertical industries	Selecting optimal edge nodes to minimize communication delays and computational overhead, and ensuring reliability.
Edge nodes deployment	Selecting optimal edge nodes to minimize communication delays and computational overhead, and ensuring reliability.
Execution of AI models on resource-constrained edge devices	Training and deploying AI models on edge devices with limited computing resources, enhancing performance and reducing latency.
Edge service management	Ensuring reliable and flexible edge service management, handling dynamic and heterogeneous IoT devices and applications.
Theoretical foundations	Establishing robust theoretical foundations for multiobjective optimization in computing, communication, storage, and energy.
Business models	Developing data-driven business models for edge computing, integrating existing cloud models and involving multiple stakeholders.

Summary

- Security and privacy protect edge devices, networks, and data using mechanisms such as **ShadowNet, TEEs, KLRA, and blockchain.**
- Service management improves reliability and reuse of decentralized services through **DEIR and middleware such as EdgeWare.**
- Deployment focuses on edge-node placement, trust, resource constraints, and integration platforms such **as EdgeX and Akraino.**
- Foundations and business use **CAP theory** and **multi-objective optimization** while enabling a shift toward data-driven edge services.

Practice

1. What are the challenges in implementing resource management in large-scale edge computing environments?
2. How do sensor attacks (camera, GNSS, LiDAR) threaten autonomous edge devices?
3. How does the CAP theorem constrain edge storage design?

Project

Explore the integration of edge computing with existing software and hardware solutions in a specific industry (e.g., manufacturing, healthcare, and 5G) and develop a prototype that demonstrates this integration. Use open-source frameworks like EdgeX Foundry (https://lfedge.org/projects/edgex-foundry/) to integrate edge computing with selected software and hardware solutions.

➢ Tasks

1. **Pick an industry** Choose a vertical & its standards — factory sensors, patient monitors, or a 5G base station.
2. **Stand up EdgeX** Deploy EdgeX Foundry; connect devices via MQTT, REST or BLE device services.
3. **Build the pipeline** Ingest → normalize → analyze at the edge; export results or events to an app or cloud.
4. **Demonstrate & assess** Show the working prototype; evaluate latency, interoperability & data protection vs. cloud-only.

➢ What to demonstrate

- **Interoperability** Heterogeneous devices & apps talking through one framework.
- **Local processing** Data ingested, normalized and analyzed at the edge.
- **Data protection** Sensitive data kept local; secured device & export services.
- **Modularity** Swap device services or add analytics without a rebuild.

Example: a smart-factory demo — EdgeX reads a temperature/vibration sensor over Modbus, flags anomalies locally, alerts a dashboard. **lfedge.org · EdgeX Foundry**

References

- Weisong Shi et al. “Edge computing: Vision and challenges”. In: IEEE Internet of Things Journal 3. 5 (2016), pp. 637–646.
- Ketan Bhardwaj, Joaquin Chung Miranda, and Ada Gavrilovska. "Towards IoT-DDoS prevention using edge computing". In: USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18). 2018.
- Shanhe Yi, Zhengrui Qin, and Qun Li. "Security and privacy issues of fog computing: A survey". In: Wireless Algorithms, Systems, and Applications (WASA). Springer. 2015, pp. 685–695.
- Zhenyu Ning et al. "Preliminary study of trusted execution environments on heterogeneous edge platforms". In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. 2018, pp. 421–426.
- Peng Li et al. "KLRA: A kernel level resource auditing tool for IoT operating system security". In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). IEEE. 2018, pp. 427–432.
- Yuxin Wang et al. “Quantitative analysis of storage requirement for autonomous vehicles”. In: Proceedings of the 16th ACM Workshop on Hot Topics in Storage and File Systems. 2024, pp. 71–78.
- Sidi Lu et al. “EdgeWare: Toward extensible and flexible middleware for connected vehicle services”. In: CCF Transactions on High Performance Computing 4. 3 (2022), pp. 339–356.
- The Linux Foundation. LF Edge: Building an Open Source Framework for the Edge. https://lfe dge.org/. Accessed: 2024-08-30. 2024.
- The Linux Foundation. Edge Foundary: An Open Edge Platform. https://www.edgexfoundry.org/. Accessed: 2026-08-29. 2026.