Towards a Digital Payment System for the Constrained Internet of Things

Mikolai Gütschow, Matthias Wählisch

{mikolai.guetschow, m.waehlisch}@tu-dresden.de

TUD Dresden University of Technology, Barkhausen Institut

Motivation and Challenges

Motivation

- Common Internet of Things (IoT) scenarios will benefit from a payment infrastructure
- Most of the IoT devices will be heterogeneous, constrained in memory, CPU, energy etc.

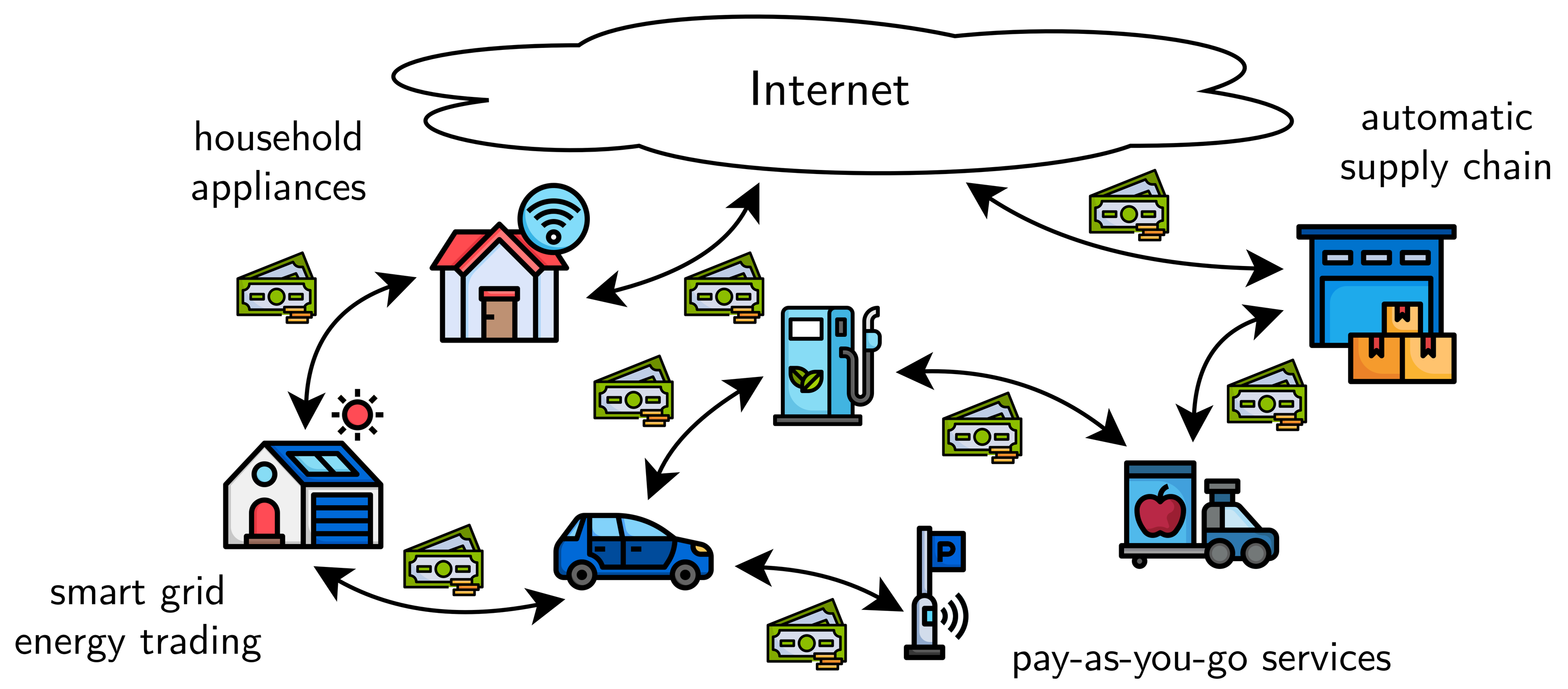
Challenges

- Storage of coins or account balances conflicts with memory or privacy
- Resource-intensive cryptographic operations challenge constrained CPUs
- Low-power wireless networking limits packet sizes and data rate

Research Questions

- Can we achieve autonomous, privacy-preserving, and scalable payments in the constrained IoT? Which limits and trade-offs exist?

Common IoT Scenarios



Limits and Potentials of Payment Approaches

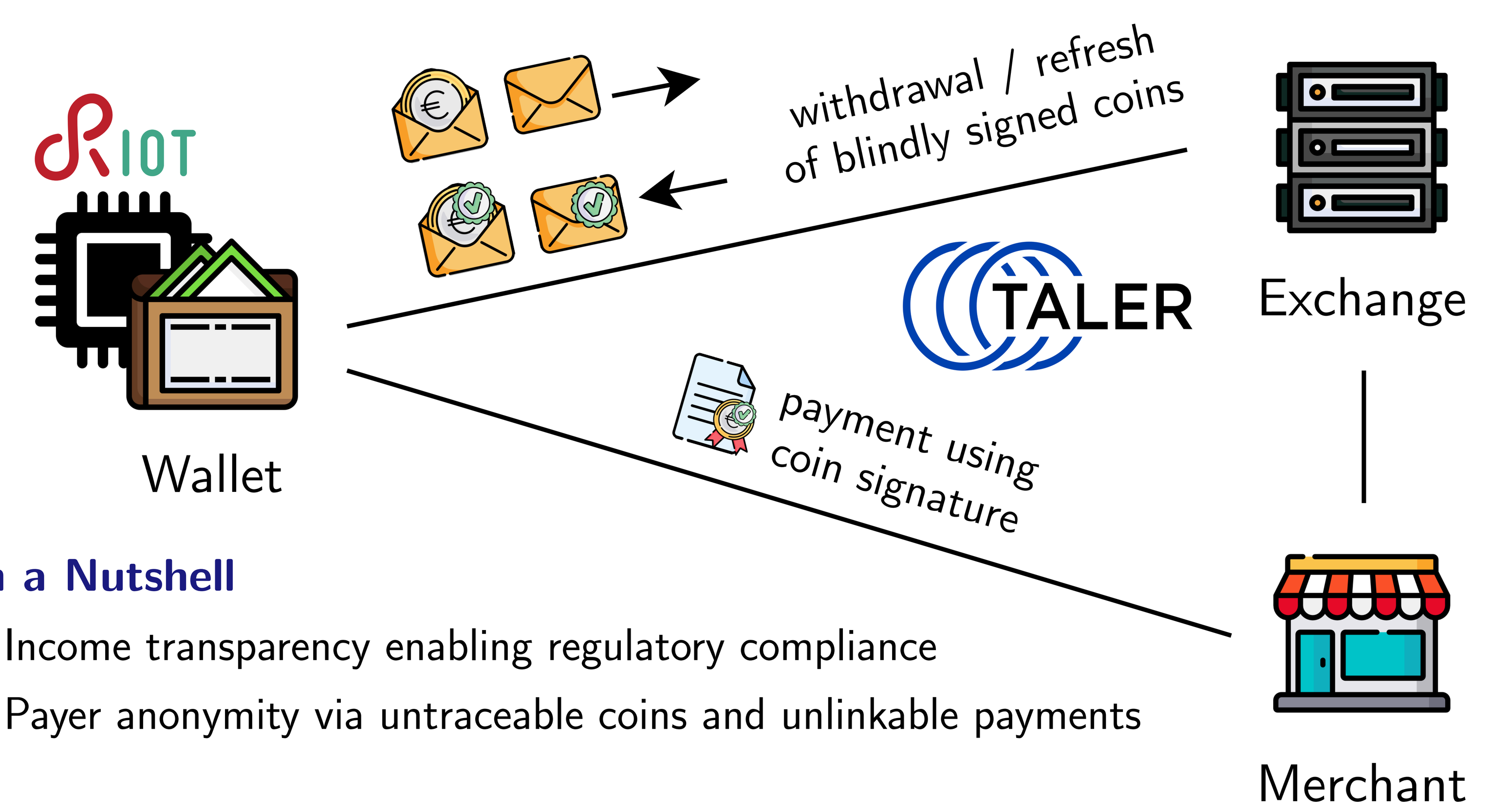
Requirements

- Autonomy for M2M-payments without human intervention
- Privacy to protect privacy-sensitive (meta-)data
- Low footprint to allow for resource-constrained devices

Comparison of Payment Approaches

	Approach		
Feature	Traditional	Cryptocurrencies	Our: E-Cash
Architecture	centralized	decentralized	(de-)centralized
Autonomy	✘	✓	✓
Privacy	✘	pseudonymity	payer
Resources	•	••••	••

GNU Taler, an E-Cash Candidate for the IoT

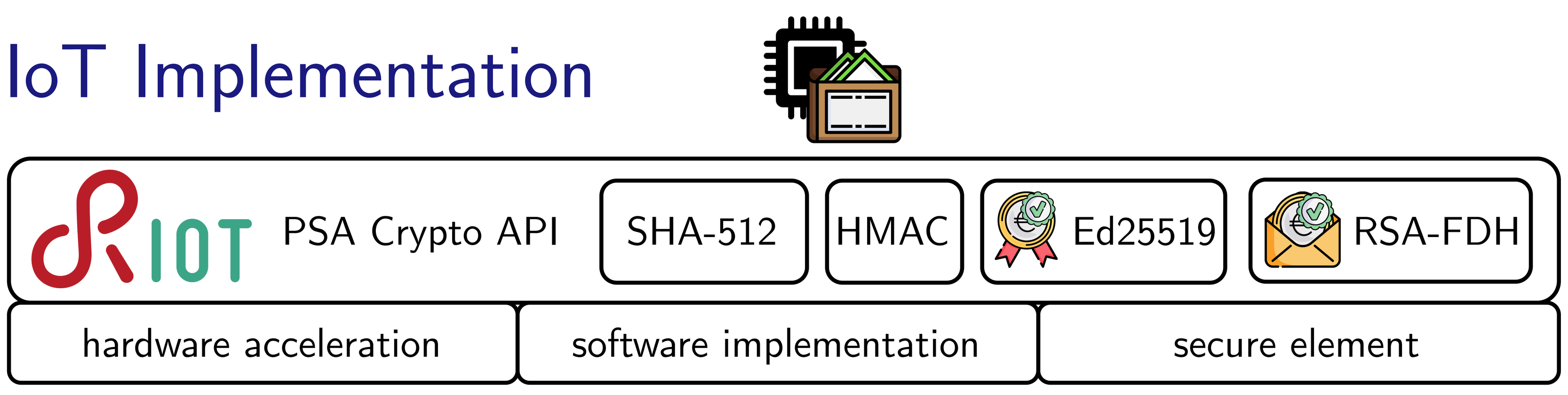


In a Nutshell

- Income transparency enabling regulatory compliance
- Payer anonymity via untraceable coins and unlinkable payments

Design of a Versatile IoT Implementation

- Our design considers high heterogeneity of IoT hardware by building on top of a general-purpose operating system providing hardware-agnostic APIs
- Our implementation is based on the backend-agnostic API for cryptographic operations in RIOT, leveraging hardware acceleration if available, with optional fallback to software

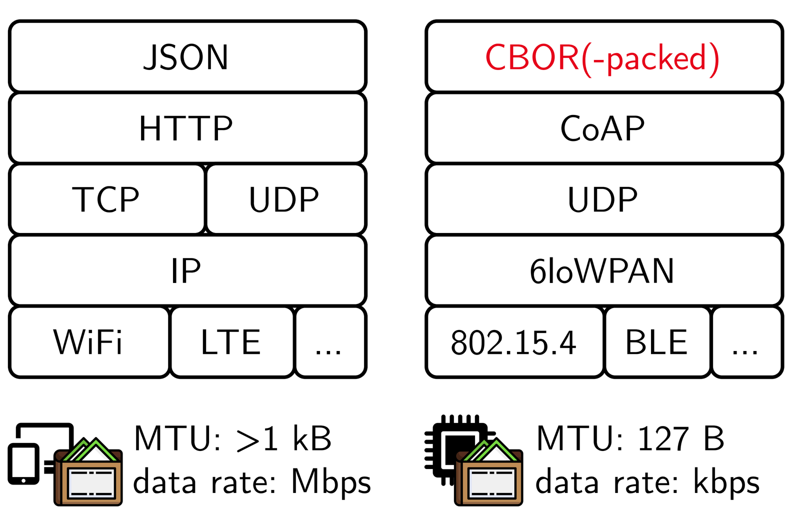


Design using a Concise Data Encoding

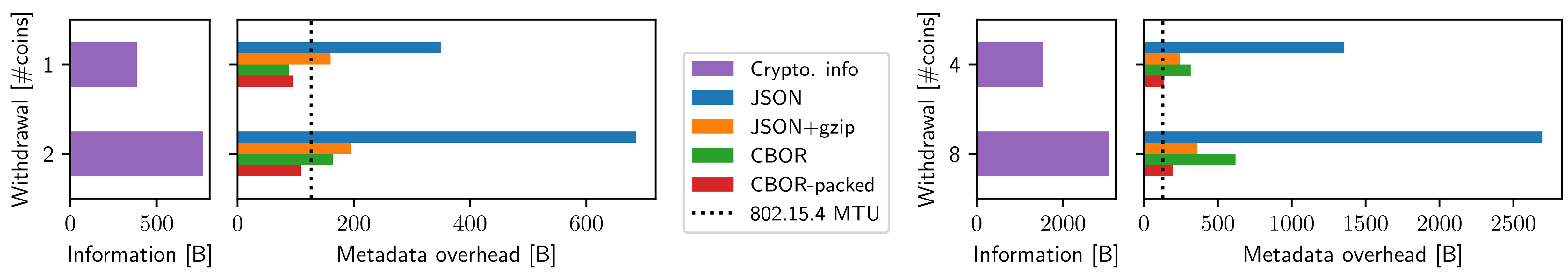
Goal: Minimize fragmentation over low-power networks

Idea: Combine suitable protocols and efficient data encoding

IoT Networking Stack

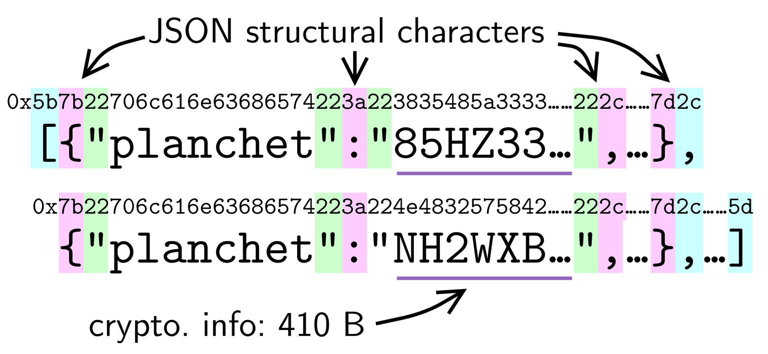


Comparison of GNU Taler Withdrawal Request Payloads

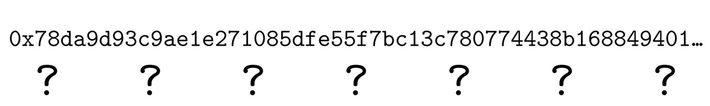


- CBOR(-packed) outperforms JSON and gzip, both in terms of resource requirements (memory, CPU, energy) and features

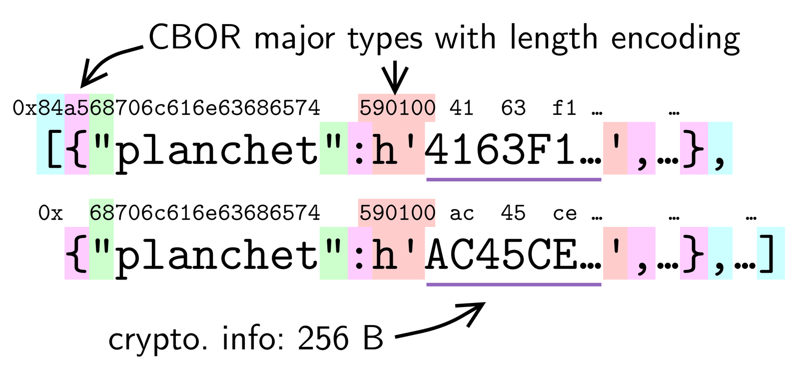
JSON, readable but verbose



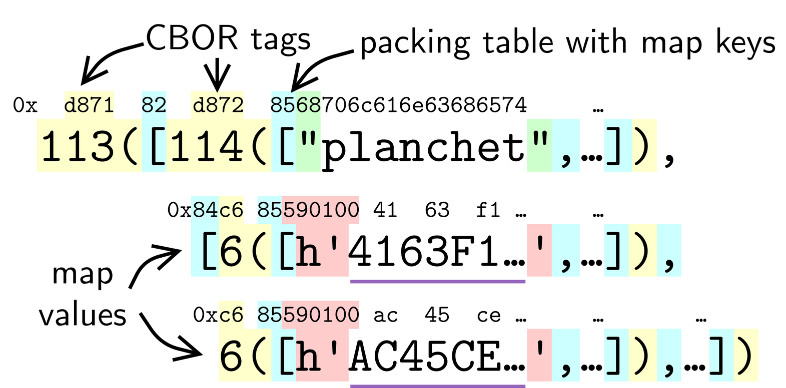
JSON+gzip, compressed but not streamable



CBOR, concise + streamable but not ASCII



CBOR-packed, less redundant



TECHNISCHE UNIVERSITÄT DRESDEN

barkhausen institut

SPONSORED BY THE Federal Ministry of Education and Research