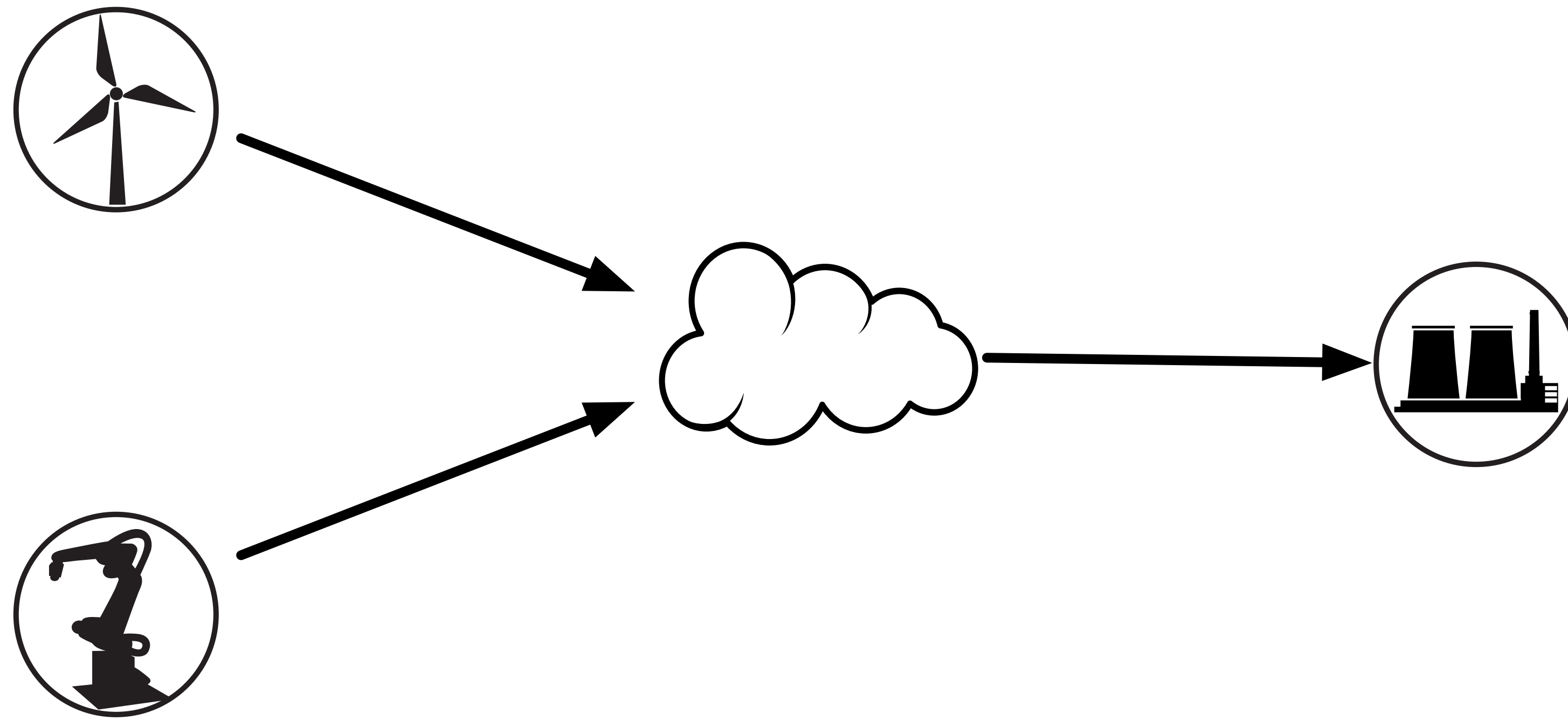




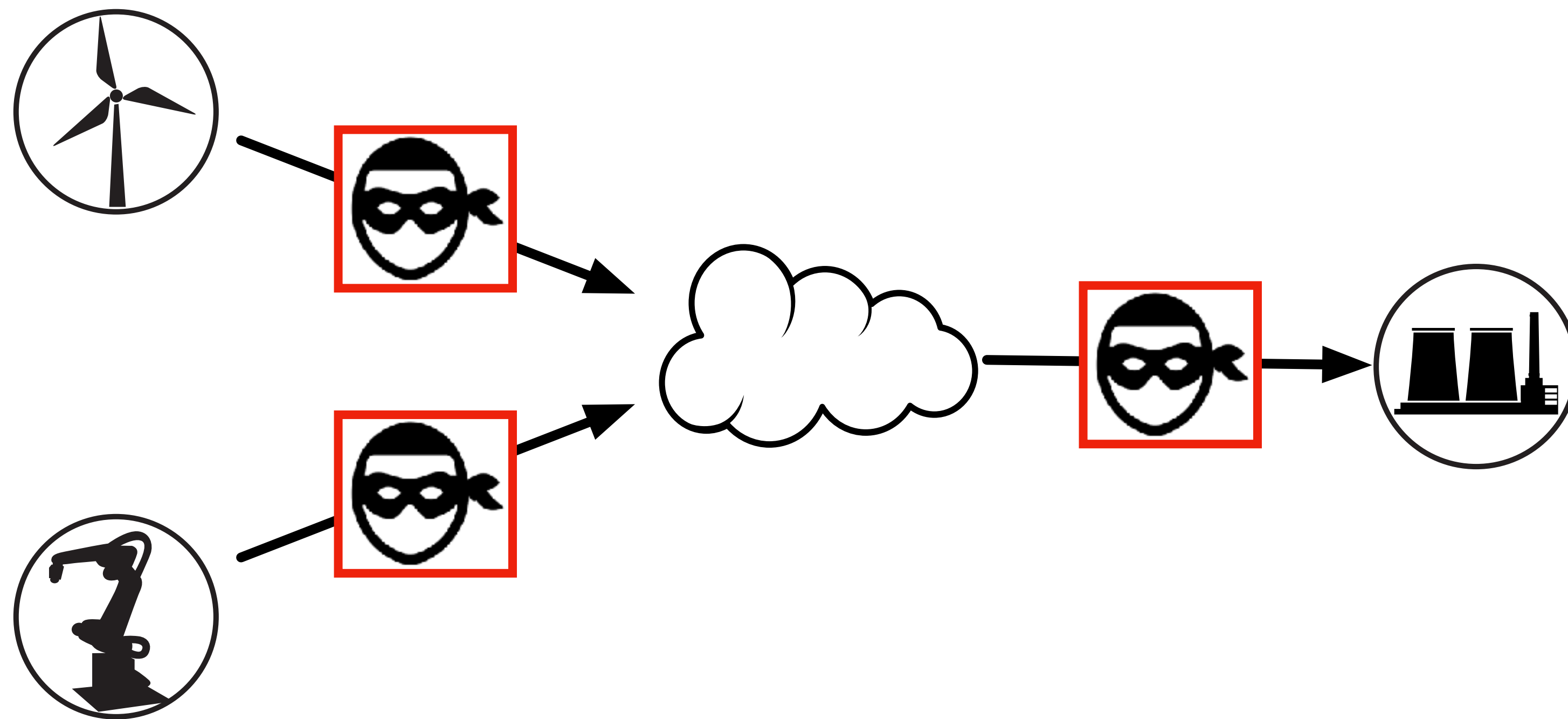
Teserakt

E4 product presentation – 20180908



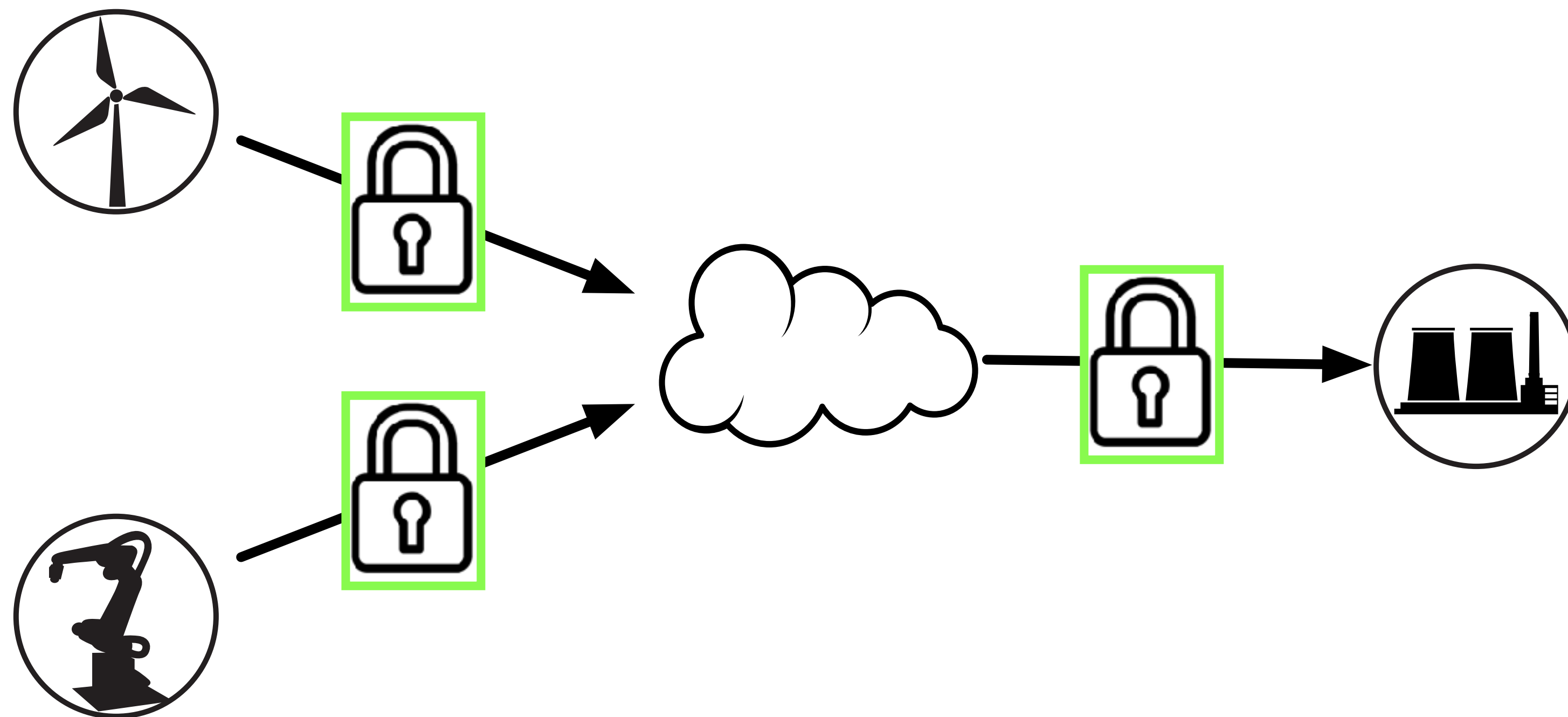
Problem

Data sent via third-party servers for reliability and performance



Problem

Data sent via third-party servers for reliability and performance
Attackers can read and corrupt data in transit

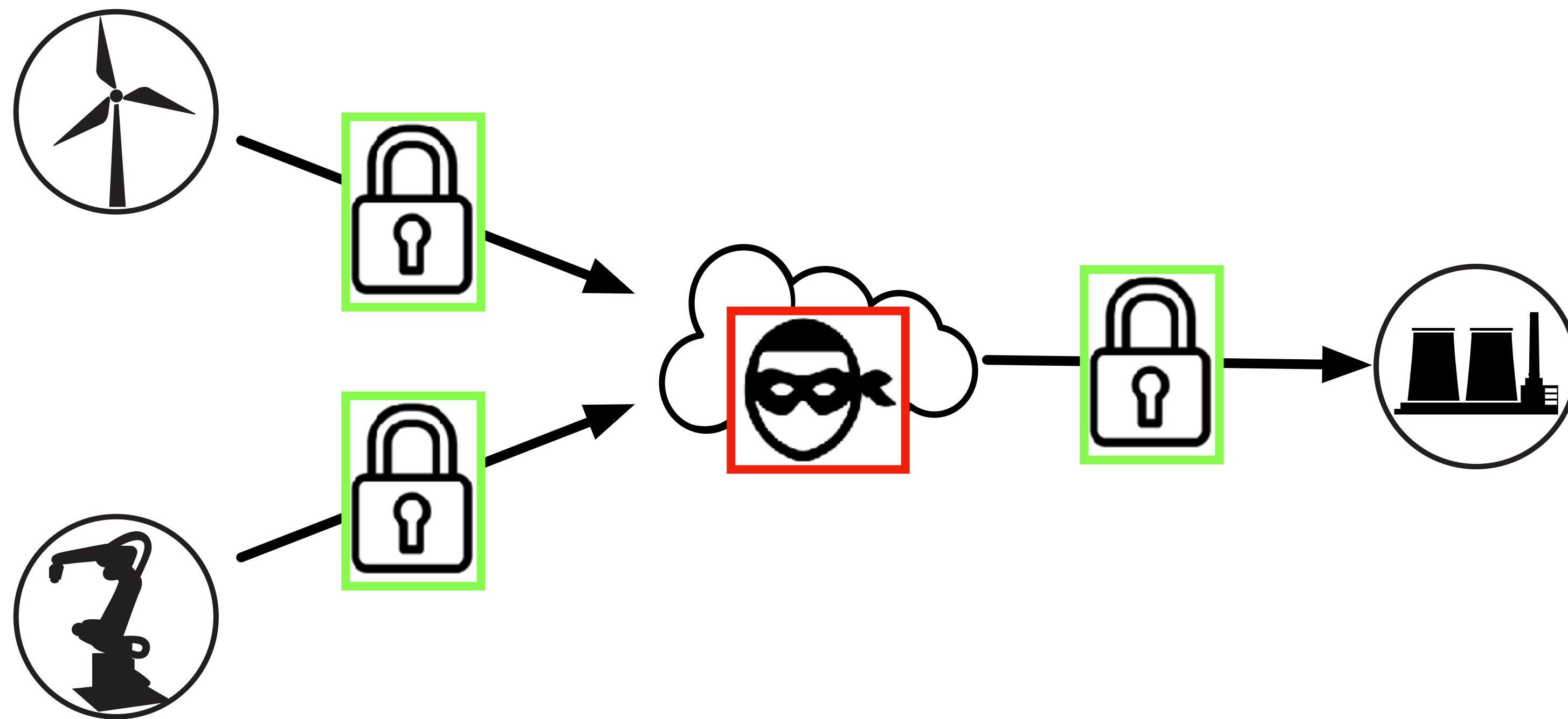


Problem

Data sent via third-party servers for reliability and performance

Attackers can read and corrupt data in transit

Client-server encryption is the default solution (TLS)



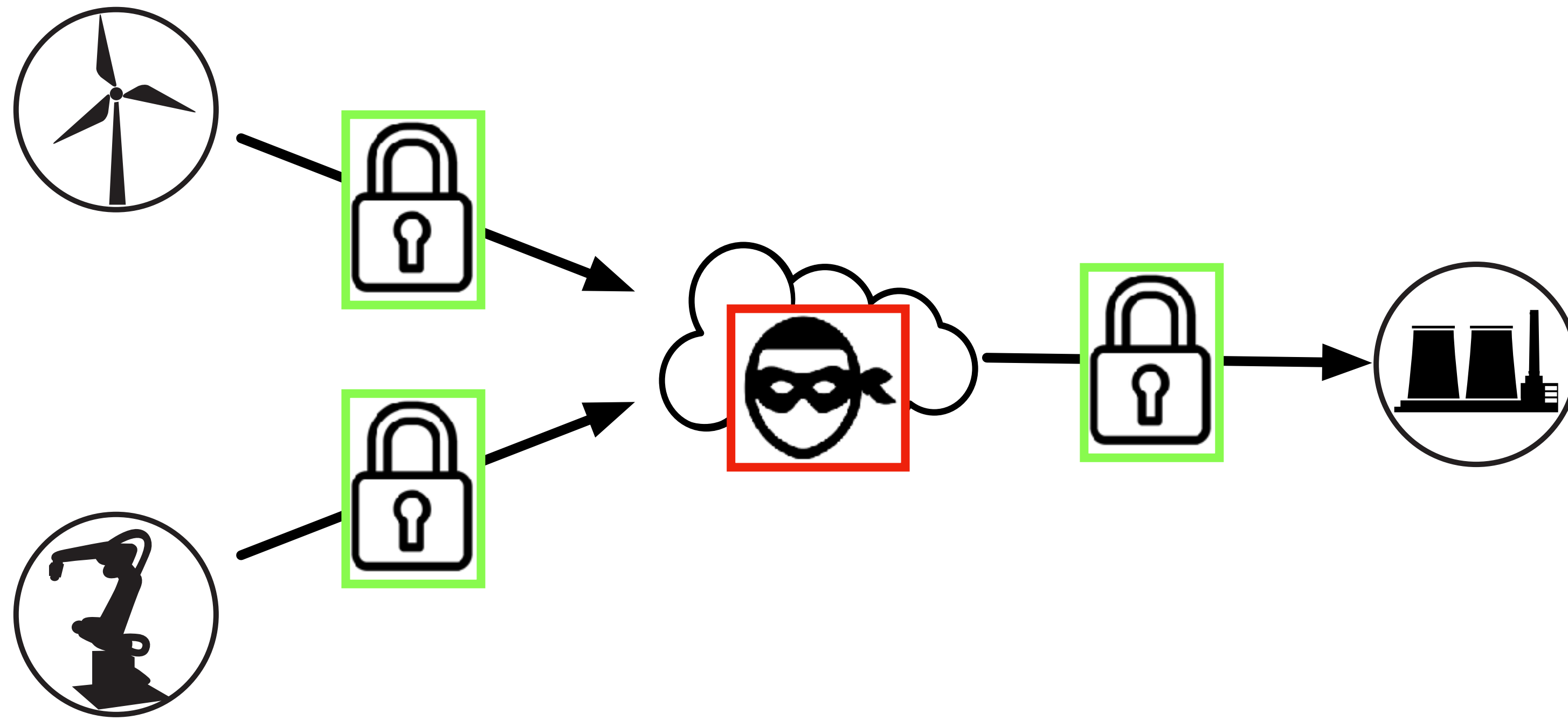
Problem

Data sent via third-party servers for reliability and performance

Attackers can read and corrupt data in transit

Client-server encryption is the default solution (TLS)

But is useless against malicious or compromised servers

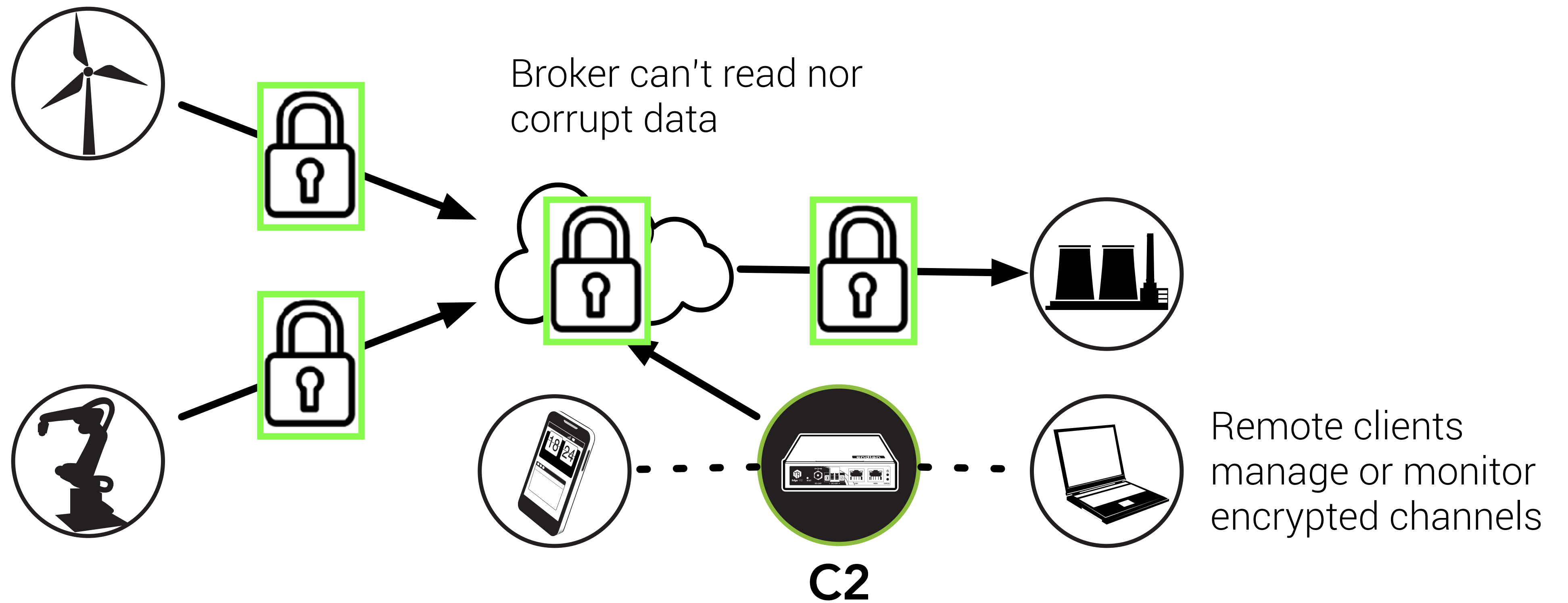


Pain points for users

No commercial solution exists

Current workaround (one global key) is cumbersome and insecure

Standard crypto won't always fit in constrained platforms (like AVR)



Our solution

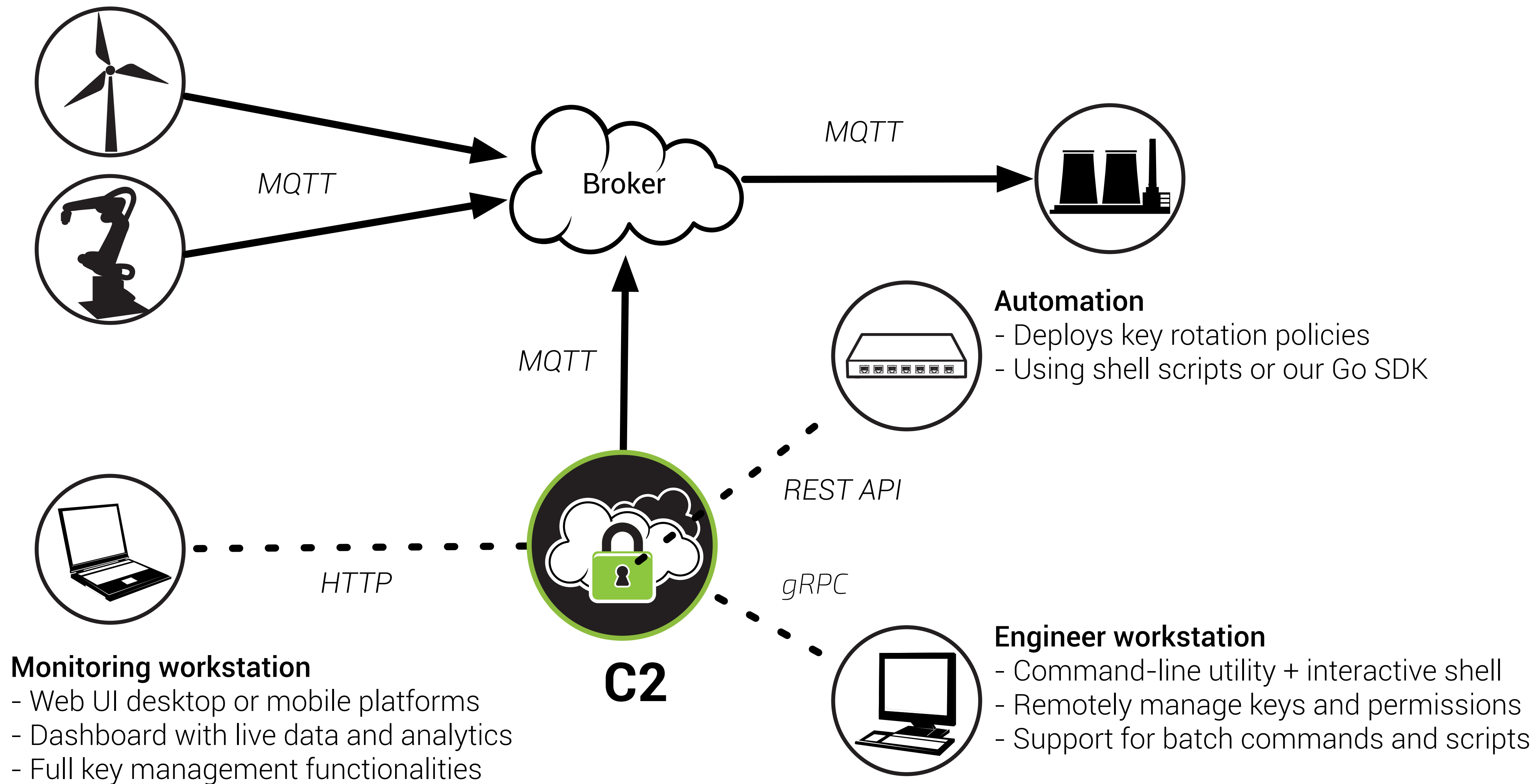
Strong encryption: confidentiality, authenticity, replay protection

Simple key management: web UI, CLI, automation of key policies

Enables advanced functionalities such as dynamic group messaging

TLS recommended, but E4 does not depend on its security

Typical deployment



Client software

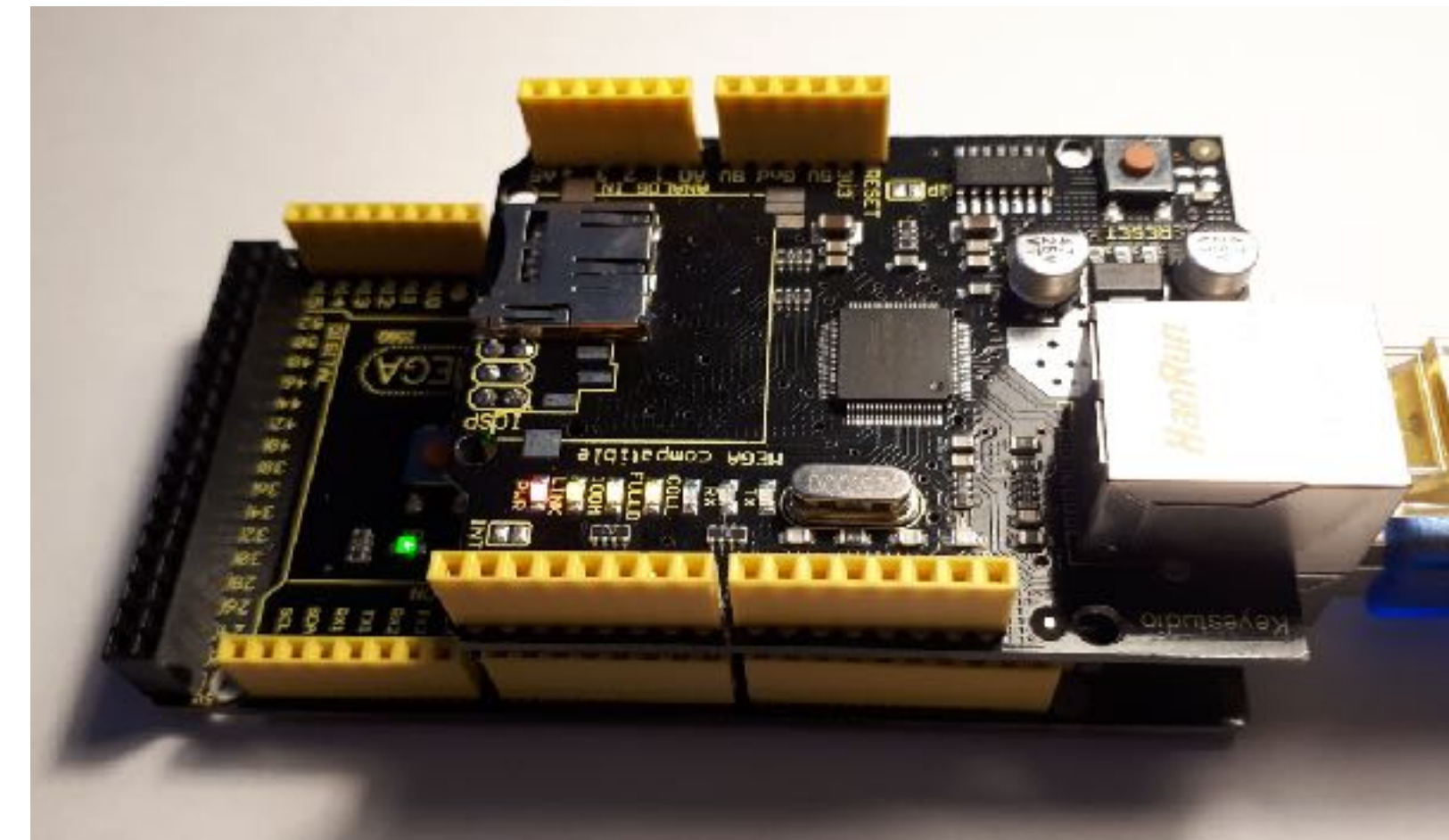
Low-memory, multiple integration options

Encrypts and decrypts MQTT payloads, with only 16 bytes overhead)

Processes key management commands receives from C2

3 options, fully interoperable:

- **C**: lightweight library tested on 8-bit AVR
- **Go**, native library for Linux/macOS/Windows
- **Java**, via our mqtt-bee fork



C2 server

Production-grade back-end, high reliability

Sends commands to clients, stores clients identities and topic keys

Written in Go, 2 interfaces:

- **HTTP/REST**: used by web UI and automation scripts
- **gRPC**: used by CLI tool and interactive shell

Soon: monitoring & analytics interface



Teserakt

*Want to try our solutions or see a demo? Any questions?
Contact us at contact@teserakt.io*