Componolit
Secure Systems Engineering

Reimplement? Reuse? Both!

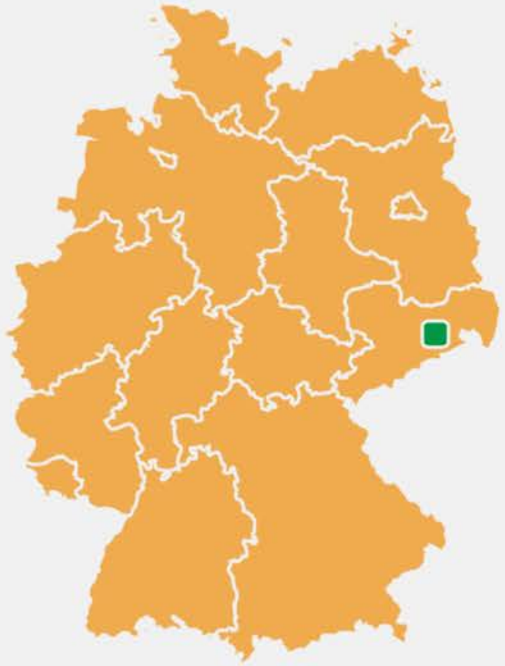
Trustworthy Systems with Genode and SPARK

Alexander Senier

Sound Static Analysis for Security Workshop
Gaithersburg, MD, June 27th, 2018

About Componolit

- **Security company** based in Dresden, Germany
- Enable customers to build secure & robust systems
  - Component-based systems
  - Program verification
- **Focus:**
  - Mobile devices
  - Industrial IoT



What is SPARK?
Language and toolset

- Programming language and tool set
- Different levels of assurance
- Adapt at your discretion

What is SPARK?
Stone level

- No side-effects in functions
- No parameter aliasing
- No pointers
- Fewer dangerous constructs

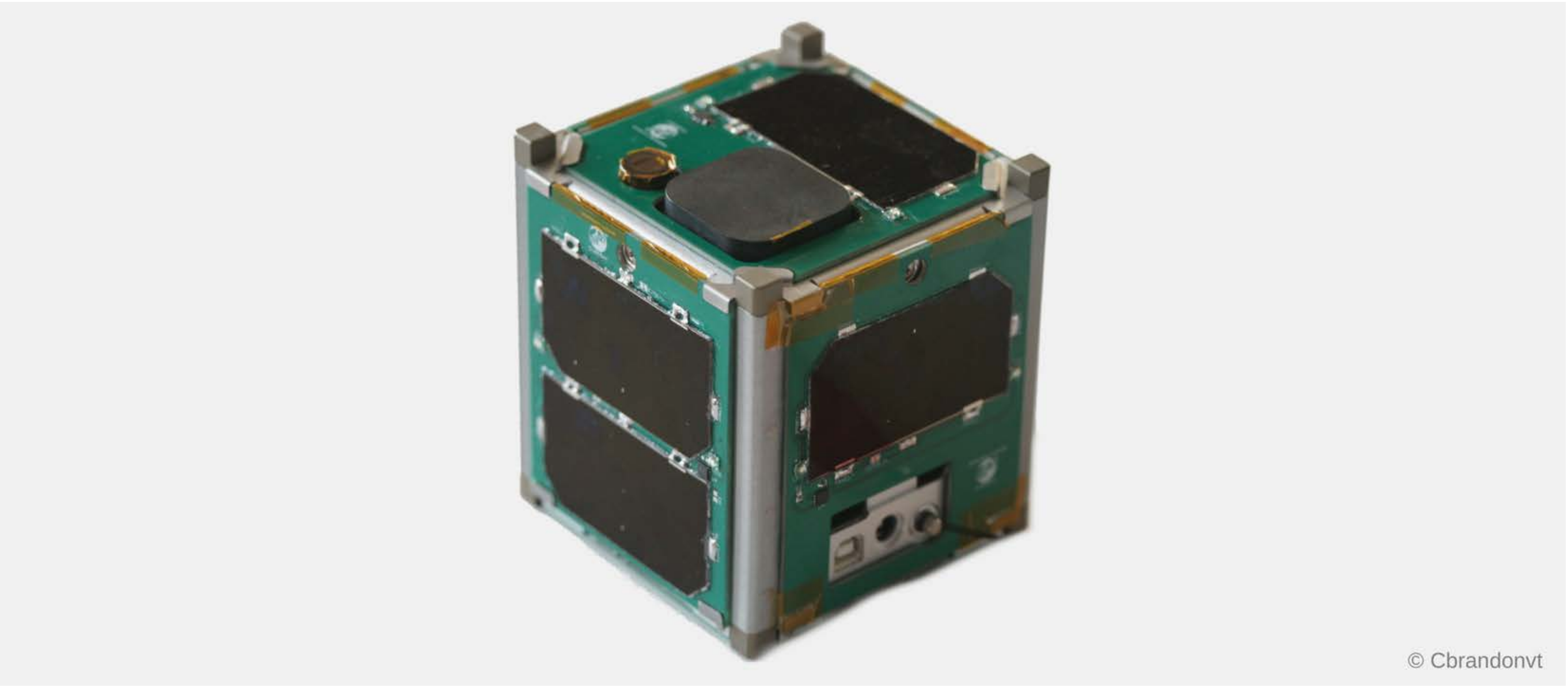
What is SPARK?
Higher assurance

- **Bronze level:** Correct initialization and data flow
- **Silver level:** Absence of runtime errors
- **Gold level:** Proof key (integrity) properties
- **Platinum level:** Functional correctness

What is SPARK?
You may know this: Jet engines

Componolit
Secure Systems Engineering

What is SPARK?
And this: Vermont Lunar CubeSat



What is SPARK?
And this: Tokeneer

What is SPARK?
And this: Muen Separation Kernel

What is SPARK?
But how about this?

Demo #1
Plain Web application

Web application
Security

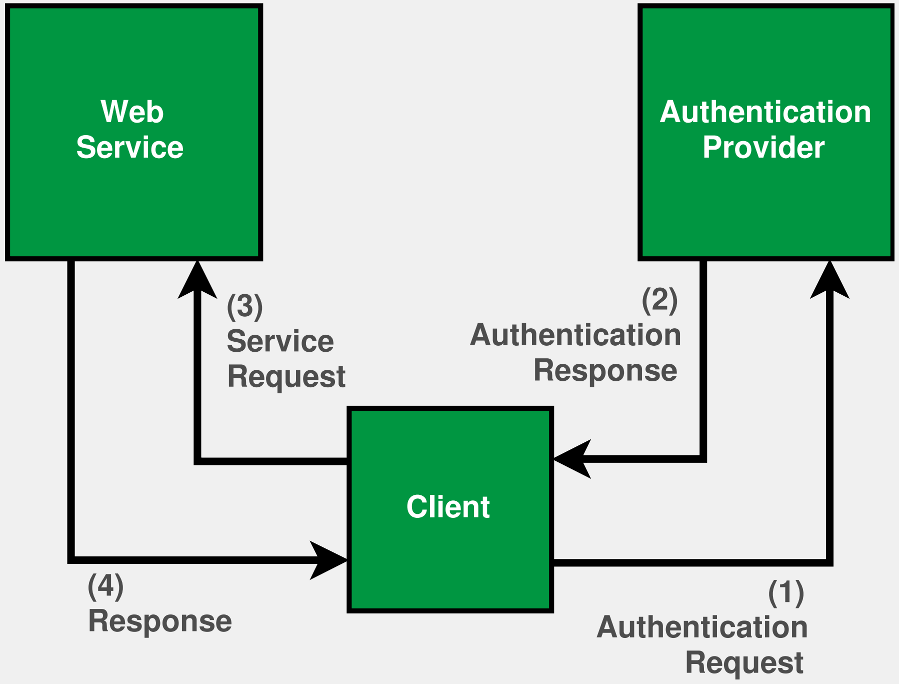
- No authentication – bad idea!
- Options
  - Passwords
  - Client certificates
  - Authentication tokens

Web application
Token-based authentication

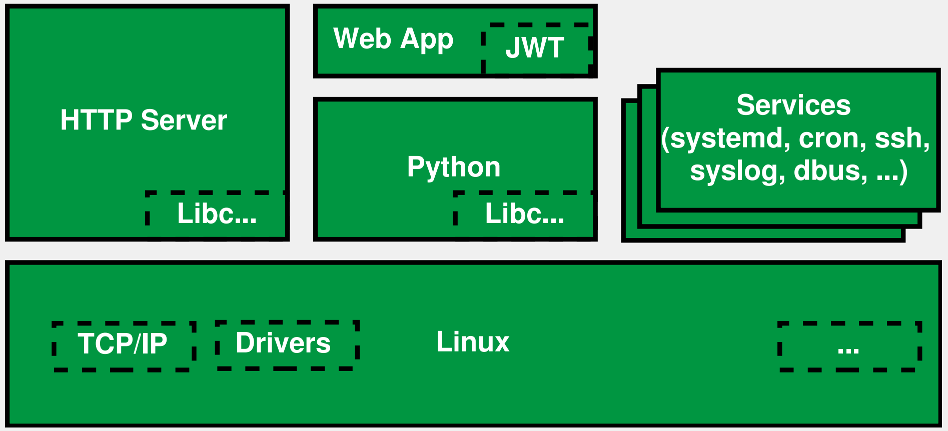
```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzd
WIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9
lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF
2QT4fwpMeJf36POk6yJV_adQssw5c
```

```
{
  "alg": "HS256",
  "typ": "JWT"
}

{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```



Token-based authentication
The monolithic approach



- A lot to trust!
- How likely is **no** critical bug within decades?
- Millions of lines of code
- Formally verifying all those components? Good luck!

Componolit
Secure Systems Engineering

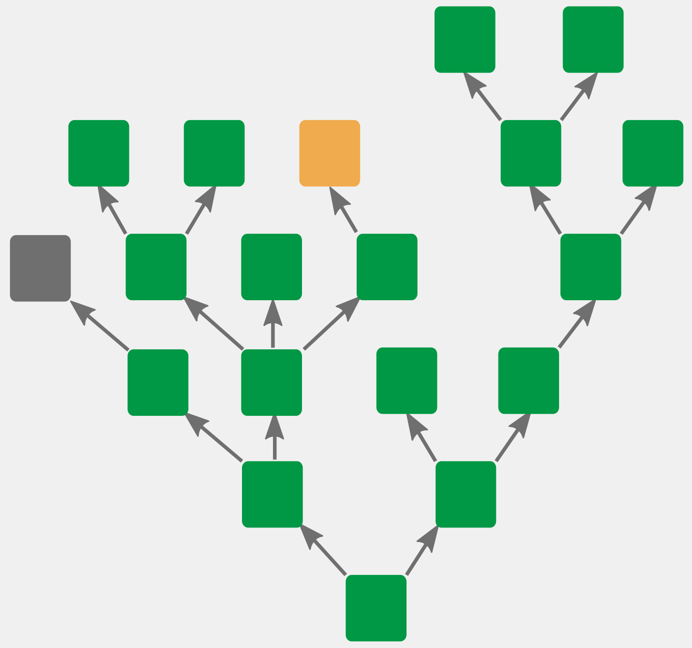
We still want trustworthy authentication for our wind turbine!

Alternatives?

Interlude
The Genode OS Framework*

- **Recursive system structure**
  - Root: Microkernel
  - Parent: Responsibility + control
  - Isolation is default
  - Strict communication policy
- **Everything is a user-process**
  - Application
  - File systems
  - Drivers, Network stacks

- **Hierarchical System Architecture**

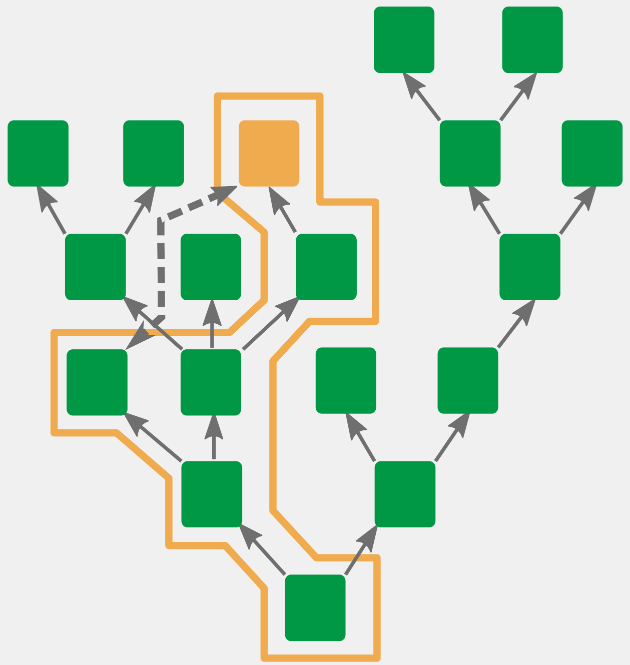


***) https://genode.org**

Interlude
Minimal Trusted Computing Base

- **Trusted Computing Base (TCB)**
  - Software required for security
  - Parents in tree
  - Services used
- **TCB reduction**
  - Application-specific
  - Example: File system
- **Sessions**

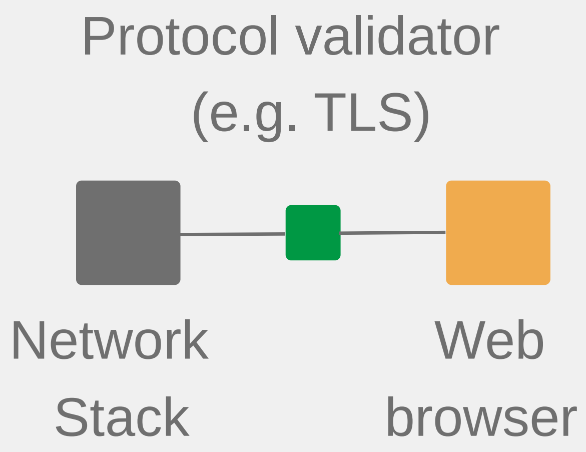
- **Per-application TCB**



Architecture for Trustworthy Systems
Strategy #1: Policy Objects

- Can't reimplement everything
- **Solution: software reuse**
  - Untrusted software (gray)
  - Policy object (green)
  - Client software (orange)
- **Policy object**
  - Establishes assumptions of client
  - Sanitizes
  - Enforces additional policies

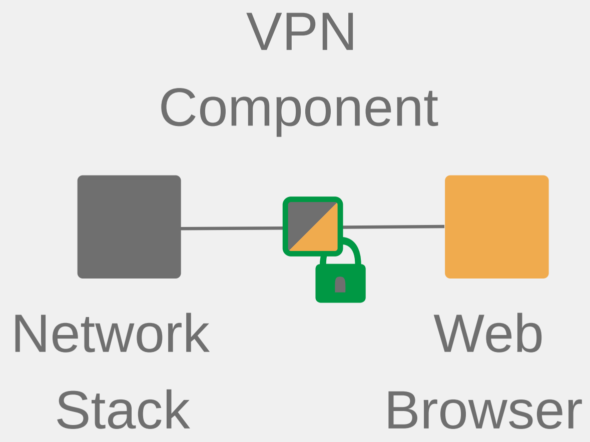
- **Policy objects**



Architecture for Trustworthy Systems
Strategy #2: Trusted Wrappers

- **Untrusted software (gray)**
  - E.g. disk, file system, cloud
- **Trusted wrapper**
  - Mandatory encryption
- **Client software (orange)**
  - No direct interaction with untrusted components
  - Minimal attack surface

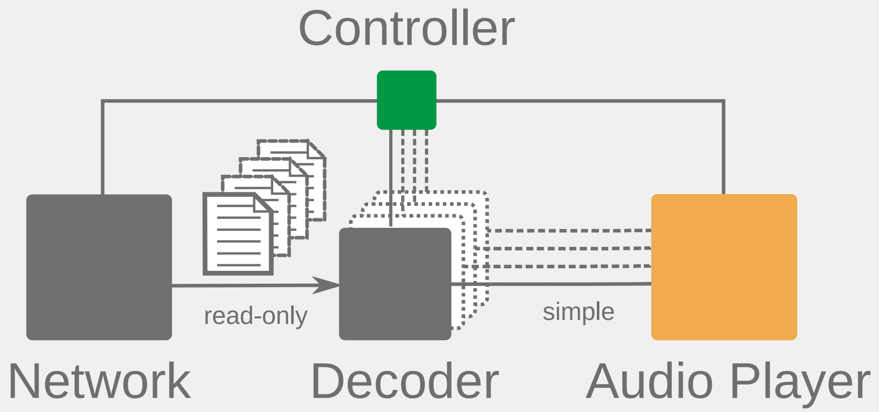
- **Trusted wrapper**



Architecture for Trustworthy Systems
Strategy #3: Transient components

- **Untrusted software**
  - E.g. Media decoder
  - No chance to get this right!
- **Transient component**
  - Temporarily instantiate untrusted software for single file/stream
  - Expose only simple interfaces (e.g. PCM audio)
  - Cleanup on completion

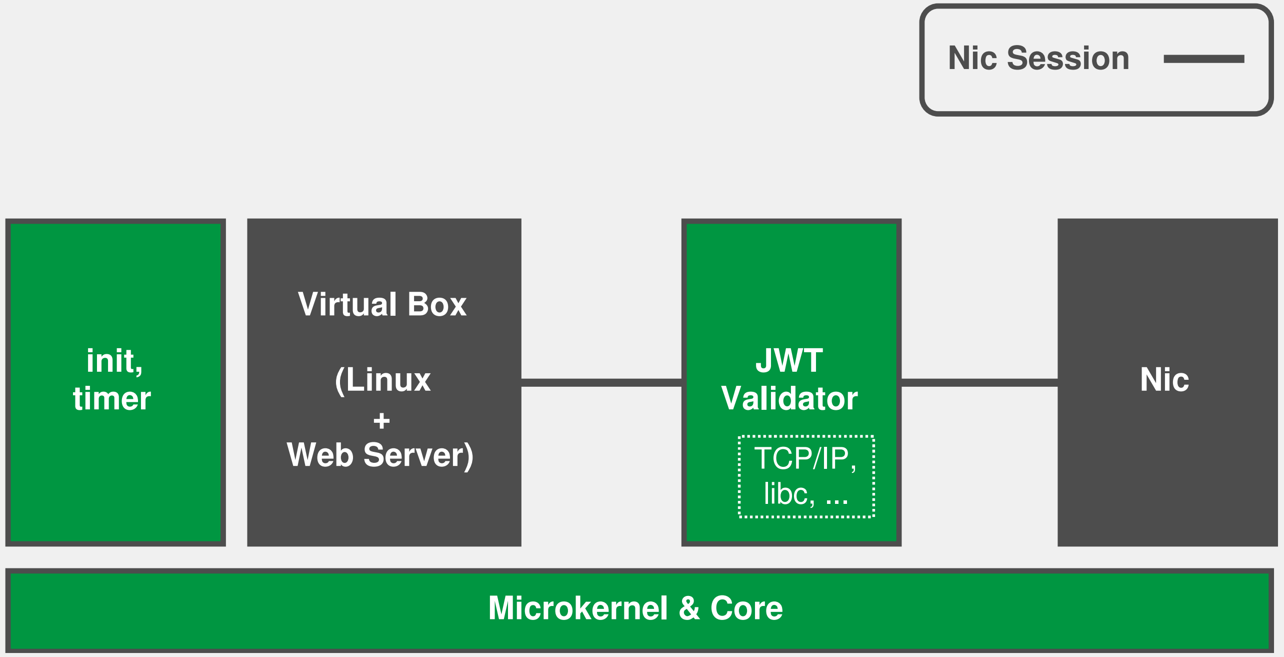
- **Transient component**



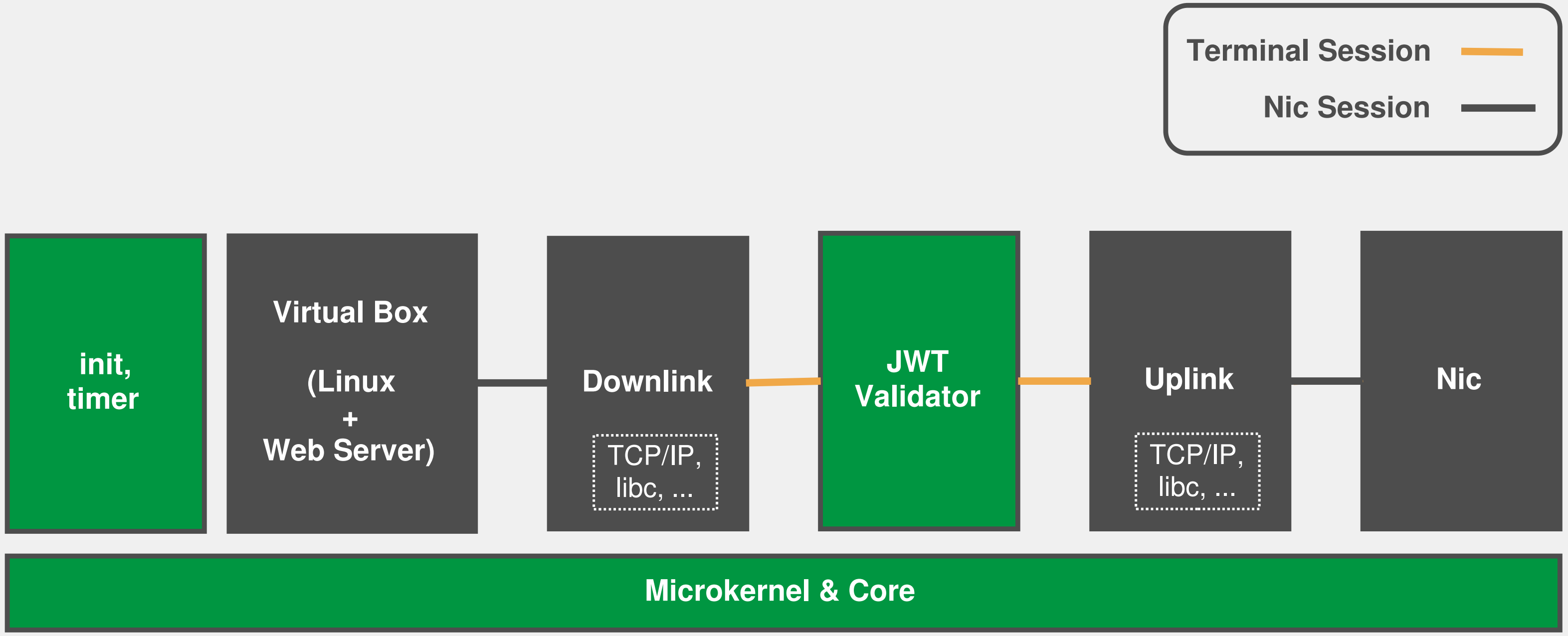
Componolit
Secure Systems Engineering

Let’s put it together.

Token-based authentication
First component-based attempt



Token-based authentication
Second component-based attempt



Demo #2
Minimal JWT Validator

Componolit
Secure Systems Engineering

Component-based architecture
Disclaimer

- Never show your authentication tokens in presentations ;-)
- Proof-of-Concept
  - No TLS in this demo!
  - Only symmetric crypto for validating JWTs for now (HMAC-SHA256)
  - Only “stone” level right now (proving absence of runtime errors TBD)
- Not a solution for availability!

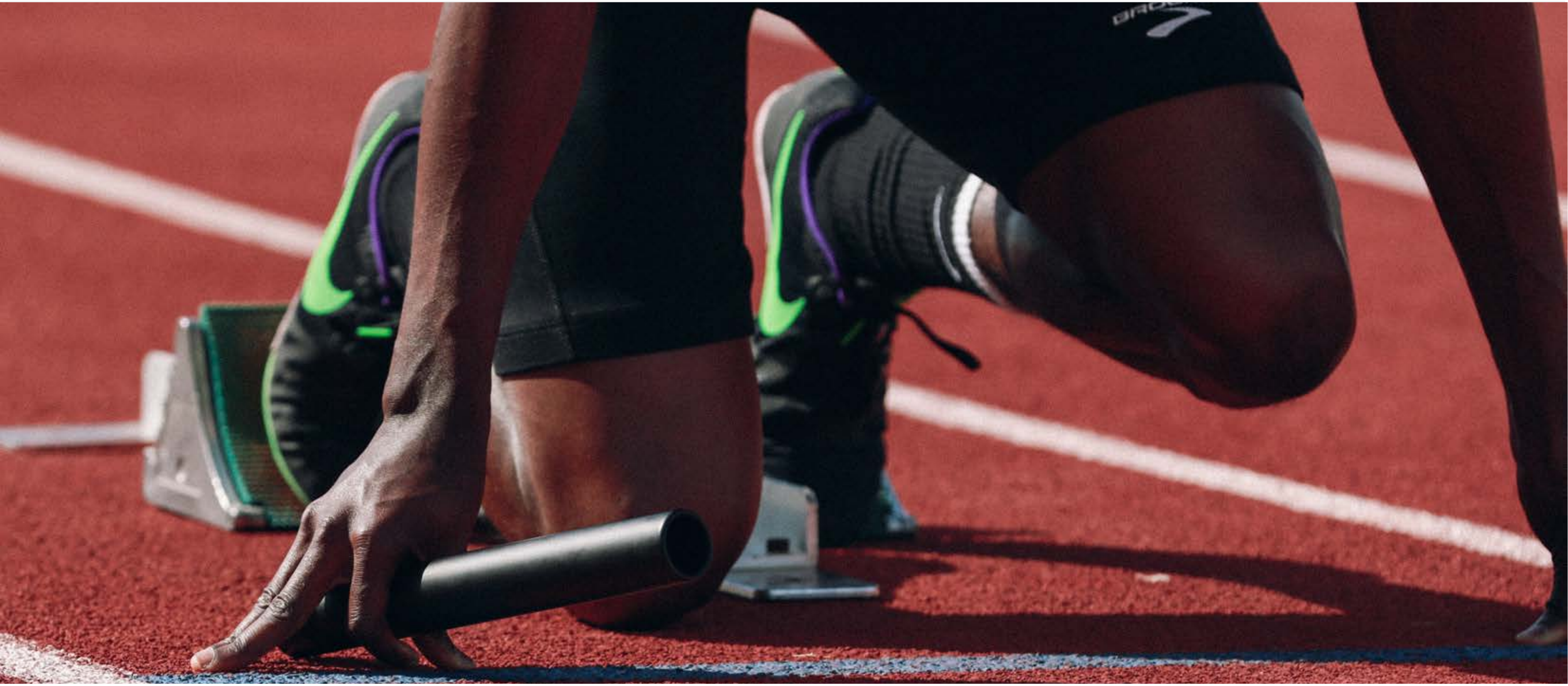
Component-based architecture
Trusted computing base

- The **TLS validator** has **3618** SLOC*:
  - Ada: 2836 (78.39%)
  - Cpp: 782 (21.61%)
- The **overall** Trusted Computing Base is **~37000** SLOC*:
  - Components: validator, microkernel, core, init, dynamic linker, RTC driver
  - cpp: 33318 (91.27%)
  - ada: 2836 (7.77%)
  - asm: 352 (0.96%)

***) generated using 'SLOCCount' by David A. Wheeler.**

Component-based architecture
But, performance?

Componolit
Secure Systems Engineering

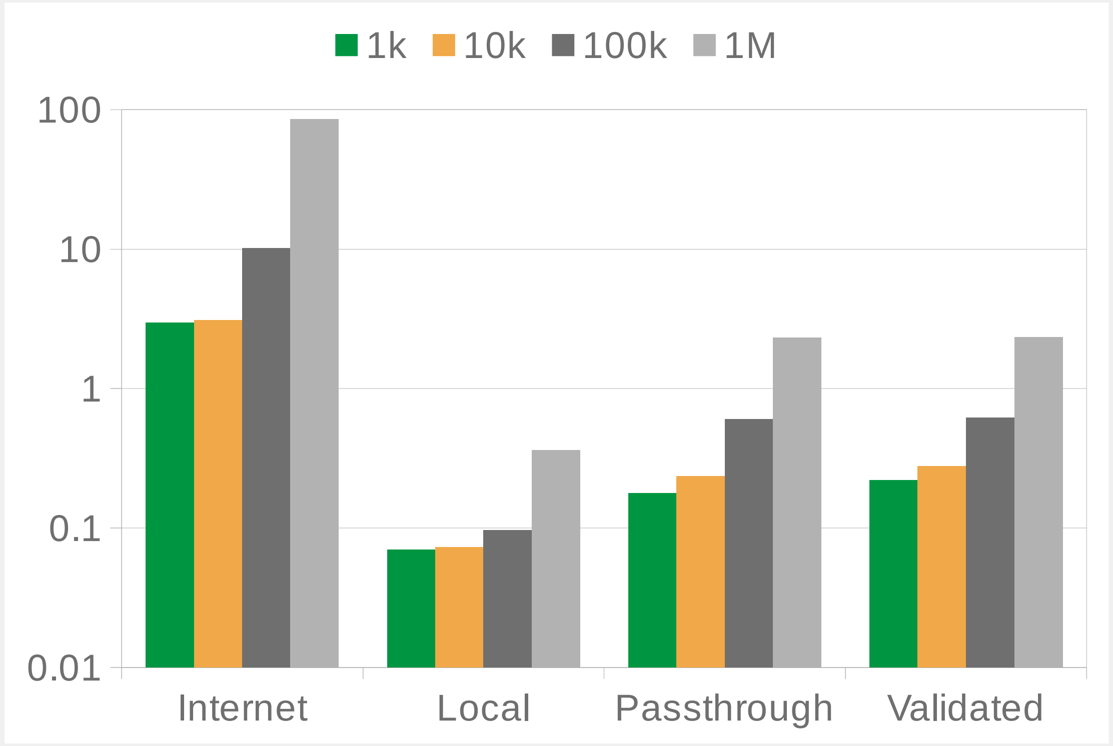


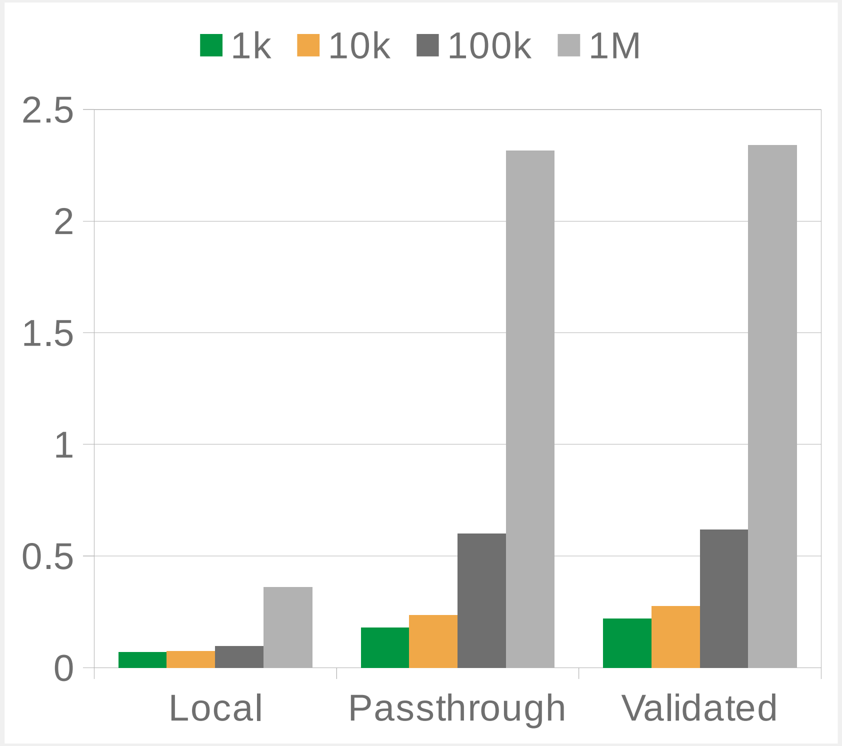
Performance Evaluation
Setup

- **Client**
  - Intel Core i5-M520, 2.4 GHz
  - Intel 82577LM GiB Ethernet
  - Debian 9.4, x86_64
  - Lighttpd 1.4.45-1
- **ab (Apache Benchmark)**
  - version 2.4.25-3+deb9u4
  - 6 concurrent requests
  - 1000 requests
    1k, 10k, 100k, 1M

- **Evaluation Setup**
  1. Internet
  2. Local webserver
  3. Local webserver through **passthrough** JWT validator
  4. Local webserver through **real** JWT validator

Performance Evaluation
Results





Mean latency between request [ms]

Component-based architecture
Reimplement? Reuse? Both!

Componolit
Secure Systems Engineering

- Component-based systems and program verification fit together very well!
- **Confidentiality & integrity**
  - No need to verify large code bases
  - Reuse of large parts of the architecture
  - Minimal trusted computing base
  - Performance: Promising, but needs evaluation in realistic setup

Component-based architecture
What else?

- Everything you saw is open source – try it!
- **JWX library for parsing JWTs (and more)**
  - `https://github.com/Componolit/jwx`
- **Demo**
  - `examples/authproxy.adb` (in JWT repository)
- **Libsparkcrypto**
  - `https://github.com/Componolit/libsparkcrypto`
- **Genode OS Framework**
  - `https://github.com/genodelabs/genode`
- **SPARK**
  - `https://www.adacore.com/download`

Questions?

Componolit
Secure Systems Engineering

Alexander Senier
Managing Director
senier@componolit.com

@Componolit · componolit.com · github.com/Componolit