CIS 4930: Secure IoT

Lecture 2

Prof. Kaushal Kafle

Fall 2024

Derived from slides by Adwait Nadkarni, William Enck

Class Notes and Clarifications

- Introductions in Canvas!
  - *~30% remaining*

- *About the readings:*
  - ***Security papers***
  - Book chapters are from textbooks so…
    - … they are not intended for paper reviews!
      - (although still good to think critically about the content)
    - You will be assigned security papers that count towards bug bounty.

Homework 1

If you are having trouble understanding the homework instructions:

- ask *specific* questions via Canvas, OR
- come to office hours to discuss!

- Asking for clarifications is always better than assuming and getting things wrong.
  - I will let you know if something is for you to figure out on your own.

- Do not forget to add the ethics statement!

Icon by freepik

Class Notes and Clarifications

- High-level Topics (and goals):
  - Basics of crypto (*this isn't a crypto course*)
  - Web and Network Security
  - IoT Security
  - Basic technologies
  - Engineering/research trade-offs
  - How to read/write/present security research papers
- Check the syllabus!
- **Note:** I reserve the right to adapt the syllabus throughout the semester… *but I will provide sufficient notice for any changes*

Non Goals

- Familiarization with the latest tools
- Professional Security Certification

Things that are not your readings

WIKIPEDIA
The Free Encyclopedia

Slashdot
News for Nerds. Stuff that matters.

digg

WEEKLY WORLD NEWS

Creature stranded on rooftop by Mississippi flood!

FBI CAPTURES BAT CHILD!

Daring agents end boy's 27-state reign of terror!

engadget

Assignment Policy

- There are 4 homework assignments
  - *Conceptual and short: Mostly question based*
  - Can discuss $\rightarrow$ Write your own answers + let me know your names.
  - 50% penalty for late homeworks within 24 hrs, 100% penalty thereafter.

Project Policy

- About groups:
  - Individual project will be allowed!
  - Keep in mind that the project deliverables will not change!
- End Result: *Research* Paper (3-5 page conference-style short paper) -> See short-paper examples in SafeThings 2022 or SafeThings 2024 (IoT Security Workshop)
- Should describe the following:
  - Introduction and Motivation
  - Experimental approach and design
  - Evaluation, result and findings

Project Policy

- Section 1: Learning IoT platform
  - **Premise**:
    - Smart home platform (any that you can program in e.g., HomeAssistant),
      - Deploying in raspberry pi (provided you have access to it) is fine
    - 3rd-party integrations/apps (e.g., HomeAssistant integrations)
    - Application Programming Interface (API),
    - Scripted Automations/routines creation using platform interface(s)
  - **Scope:**
    - Demonstrate applied knowledge
    - Don't try to learn some new *non-security field*
      - *(e.g., if you do not know NLP/LLM then…)*
    - Be realistic about what can be accomplished in a single semester.
    - However, the work should reflect real thought and effort.

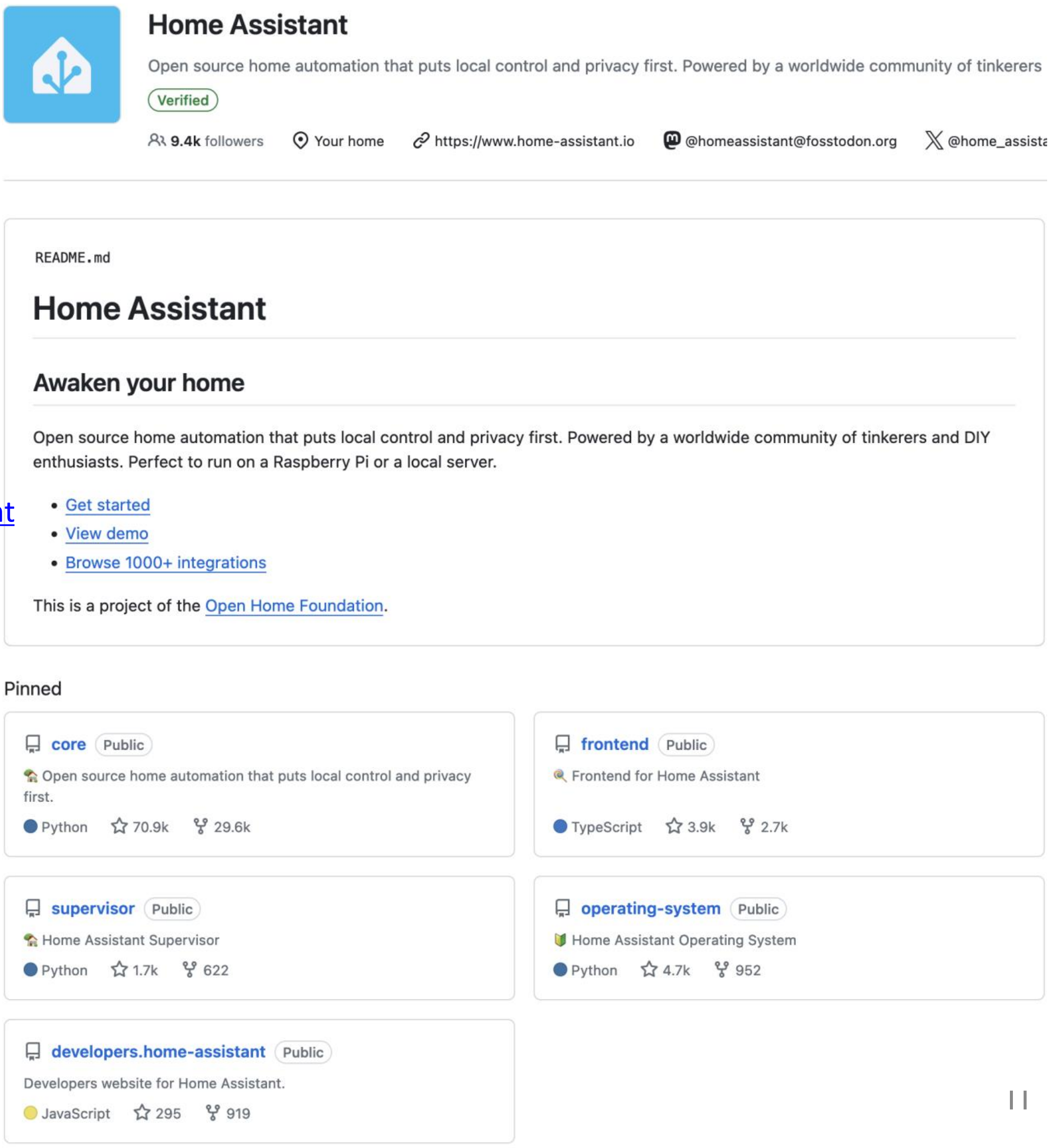
Project Policy

- *Create a Project Proposal:*
  - Outline team members clearly (name, email and U#)
  - Create an ordered list of projects
    - Propose at least 3 unique projects in order of interest (with a short description and scope)
- *Meet with me (Office Hours + by appointment)*
  - I have the end say on your project and group
    - Hopefully, I can resolve the constraints implied
    - One group per project
  - Upload project proposal to Canvas by 09/05
- The grade will be based on the following factors: *novelty*, *depth*, *correctness*, *clarity of presentation*, and ***effort***.

HomeAssistant

https://github.com/home-assistant

Have you used open-source tools before?



Home Assistant

Open source home automation that puts local control and privacy first. Powered by a worldwide community of tinkerers

Verified

9.4k followers · Your home · https://www.home-assistant.io · @homeassistant@fosstodon.org · @home_assista

README.md

Home Assistant

Awaken your home

Open source home automation that puts local control and privacy first. Powered by a worldwide community of tinkerers and DIY enthusiasts. Perfect to run on a Raspberry Pi or a local server.

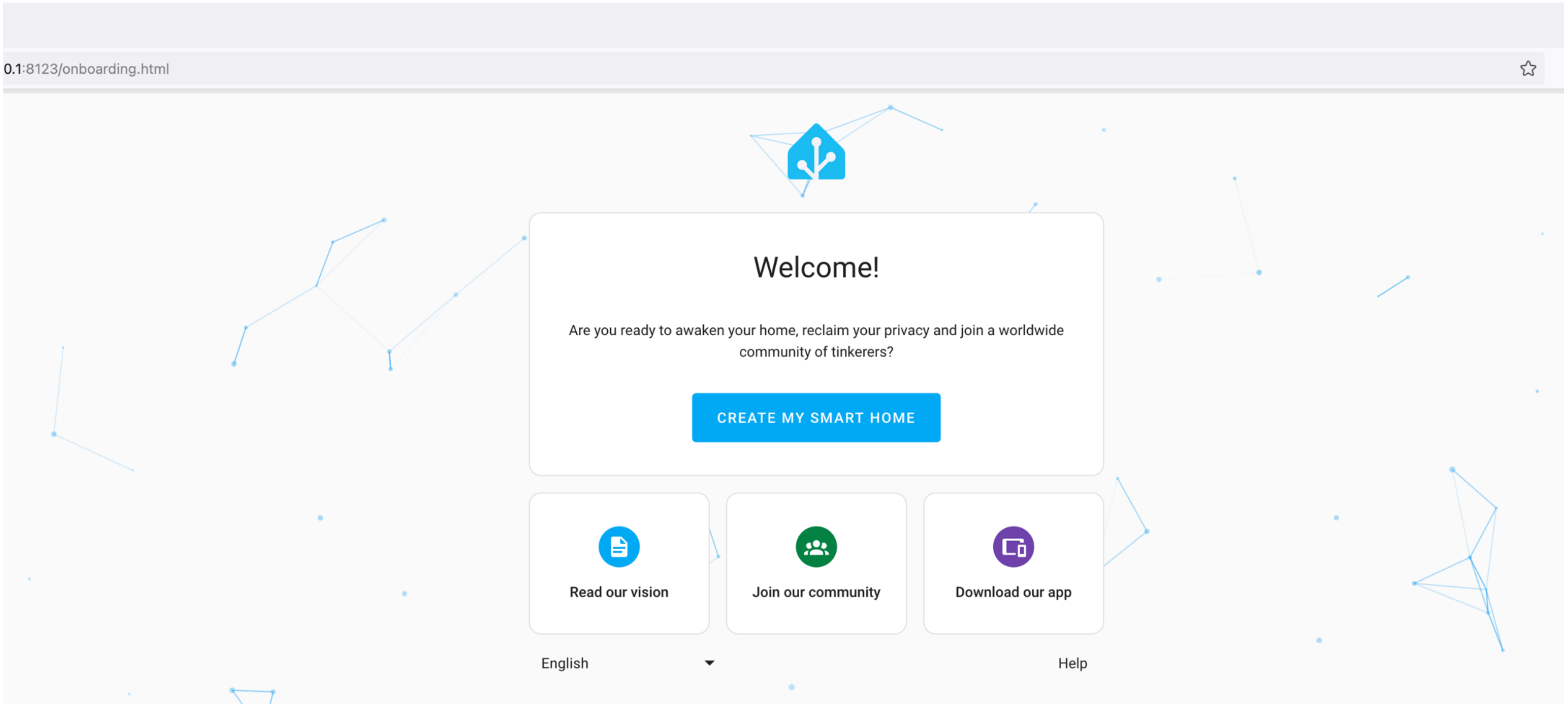
- Get started
- View demo
- Browse 1000+ integrations

This is a project of the Open Home Foundation.

Pinned

Repository	Visibility	Description	Language	Stars	Forks
core	Public	Open source home automation that puts local control and privacy first.	Python	70.9k	29.6k
frontend	Public	Frontend for Home Assistant	TypeScript	3.9k	2.7k
supervisor	Public	Home Assistant Supervisor	Python	1.7k	622
operating-system	Public	Home Assistant Operating System	Python	4.7k	952
developers.home-assistant	Public	Developers website for Home Assistant.	JavaScript	295	919

HomeAssistant

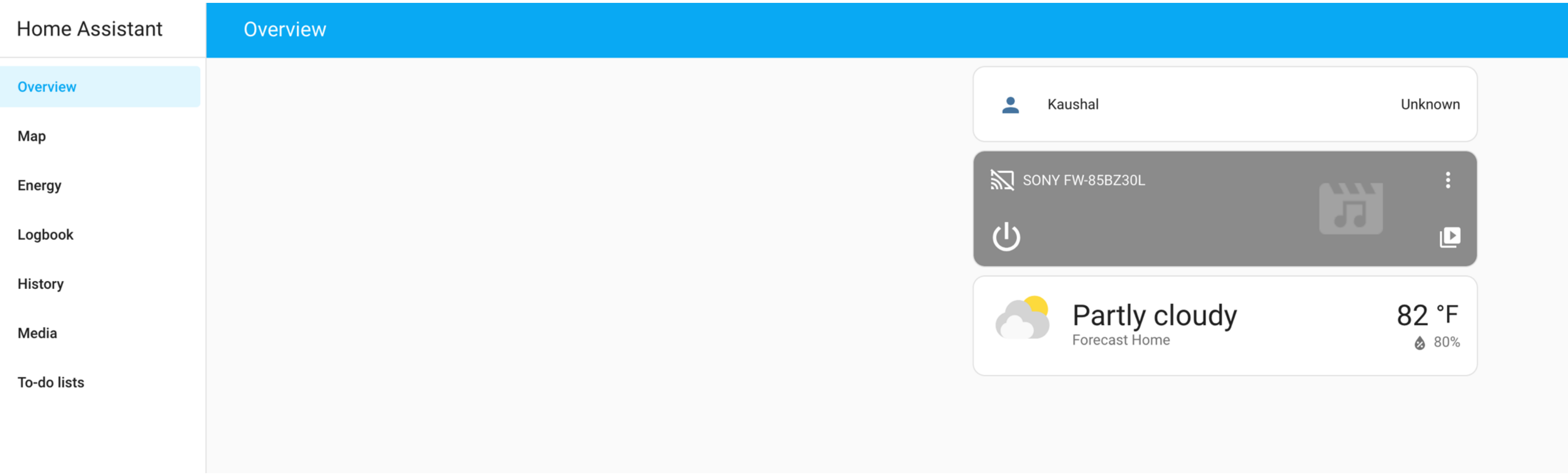


Welcome!

Are you ready to awaken your home, reclaim your privacy and join a worldwide community of tinkerers?

After installation..

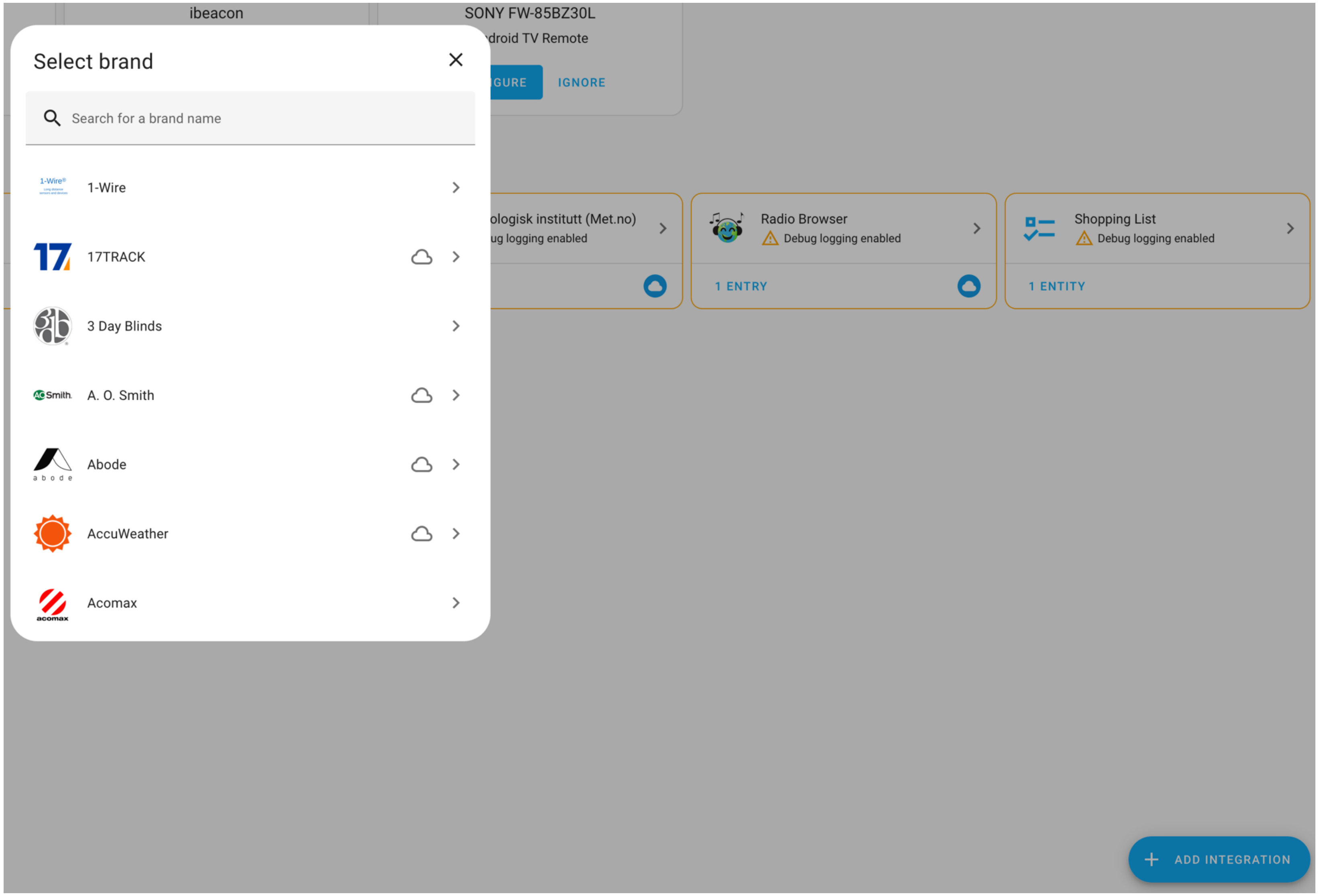
HomeAssistant



Dashboard (initially empty)..

HomeAssistant

Can add integrations!



ibeacon

SONY FW-85BZ30L
droid TV Remote
GURE IGNORE

Select brand

Search for a brand name

- 1-Wire
- 17TRACK
- 3 Day Blinds
- A. O. Smith
- Abode
- AccuWeather
- Acomax

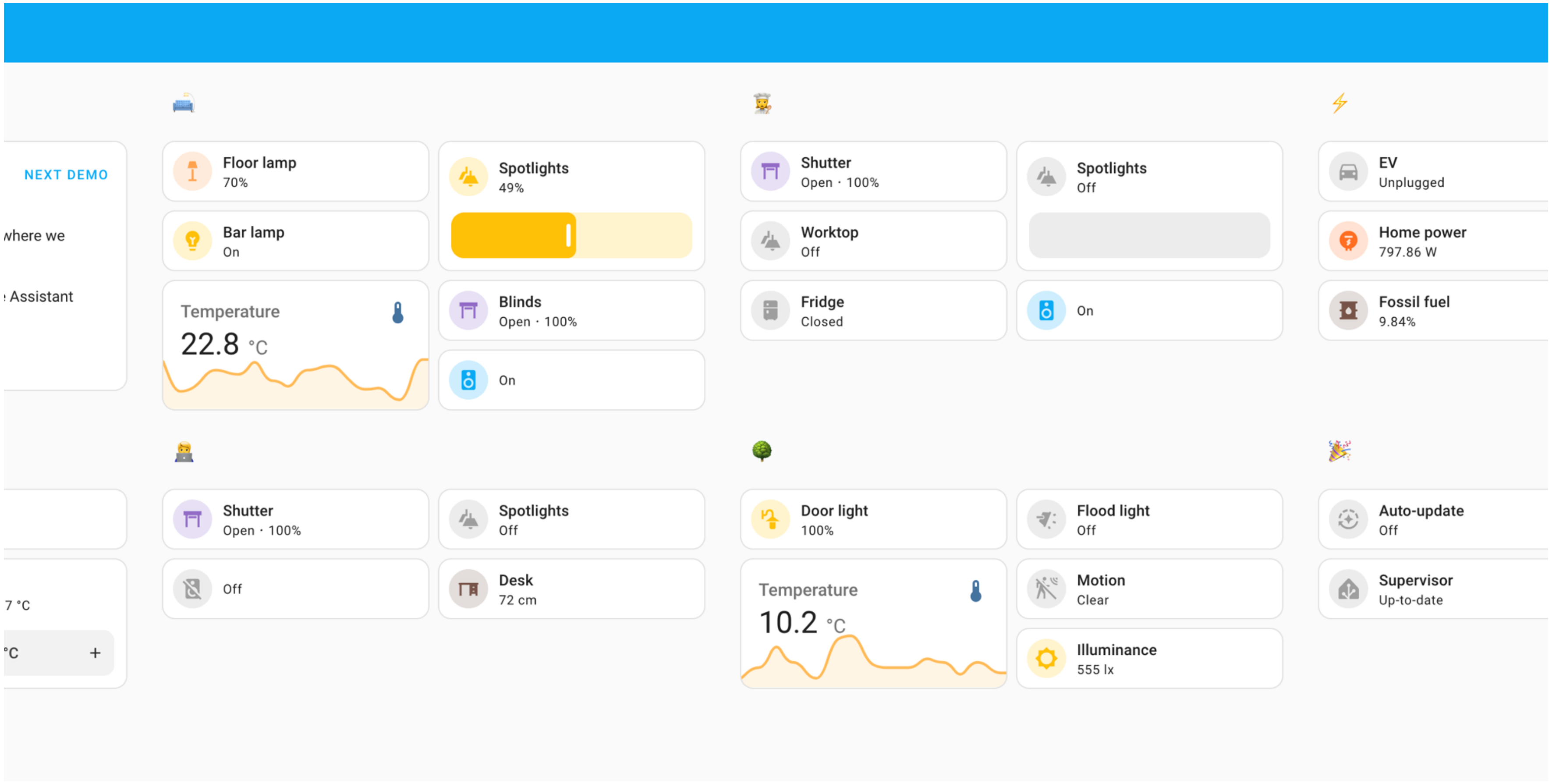
ologisk institutt (Met.no)
ug logging enabled

Radio Browser
Debug logging enabled
1 ENTRY

Shopping List
Debug logging enabled
1 ENTITY

ADD INTEGRATION

HomeAssistant



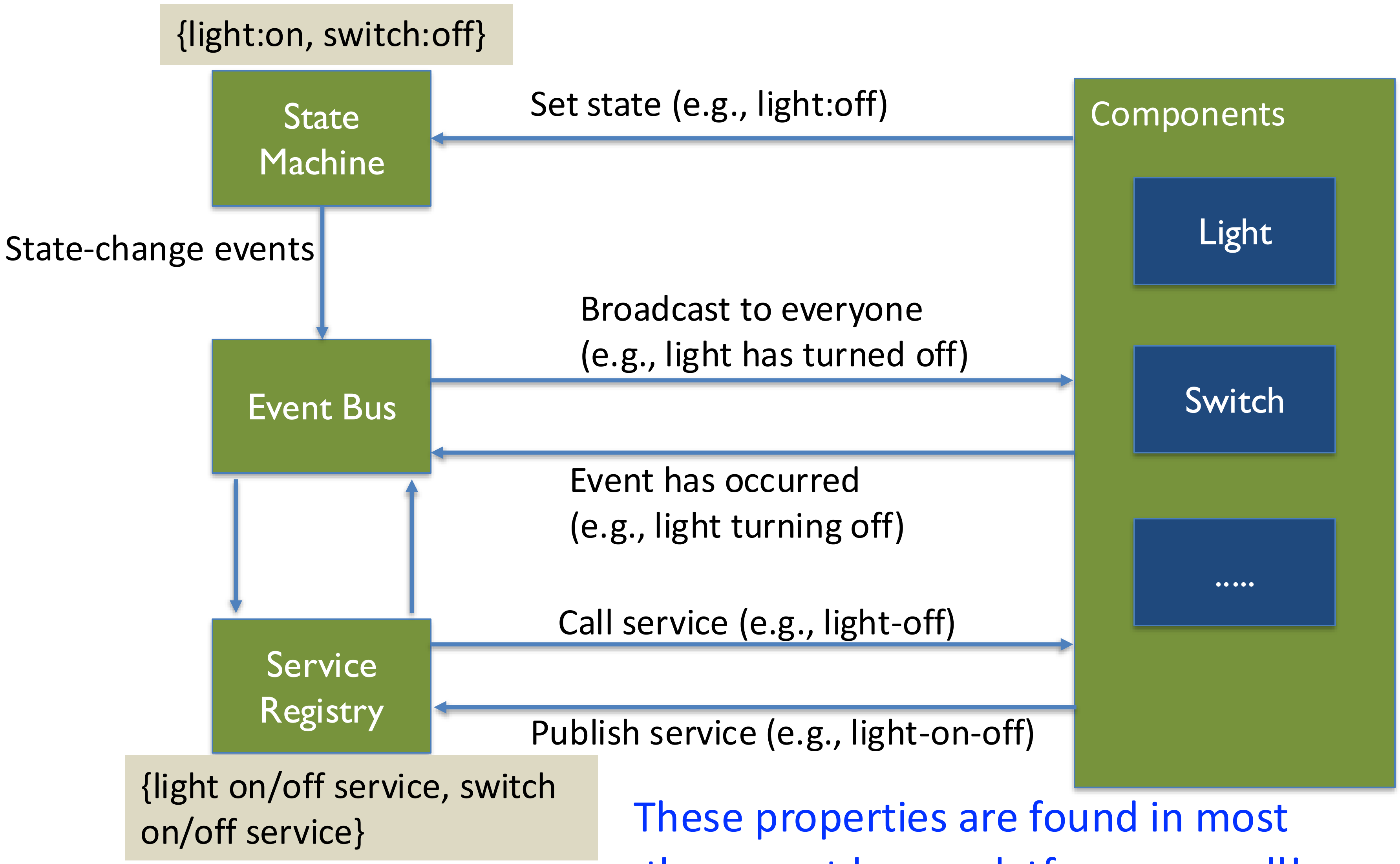
Central Dashboard (sample)..

Features

- Open-source
- Allows completely local deployments!
- Can run on raspberry-pi or virtualbox (if you want)
- Central dashboard where you can monitor your home!

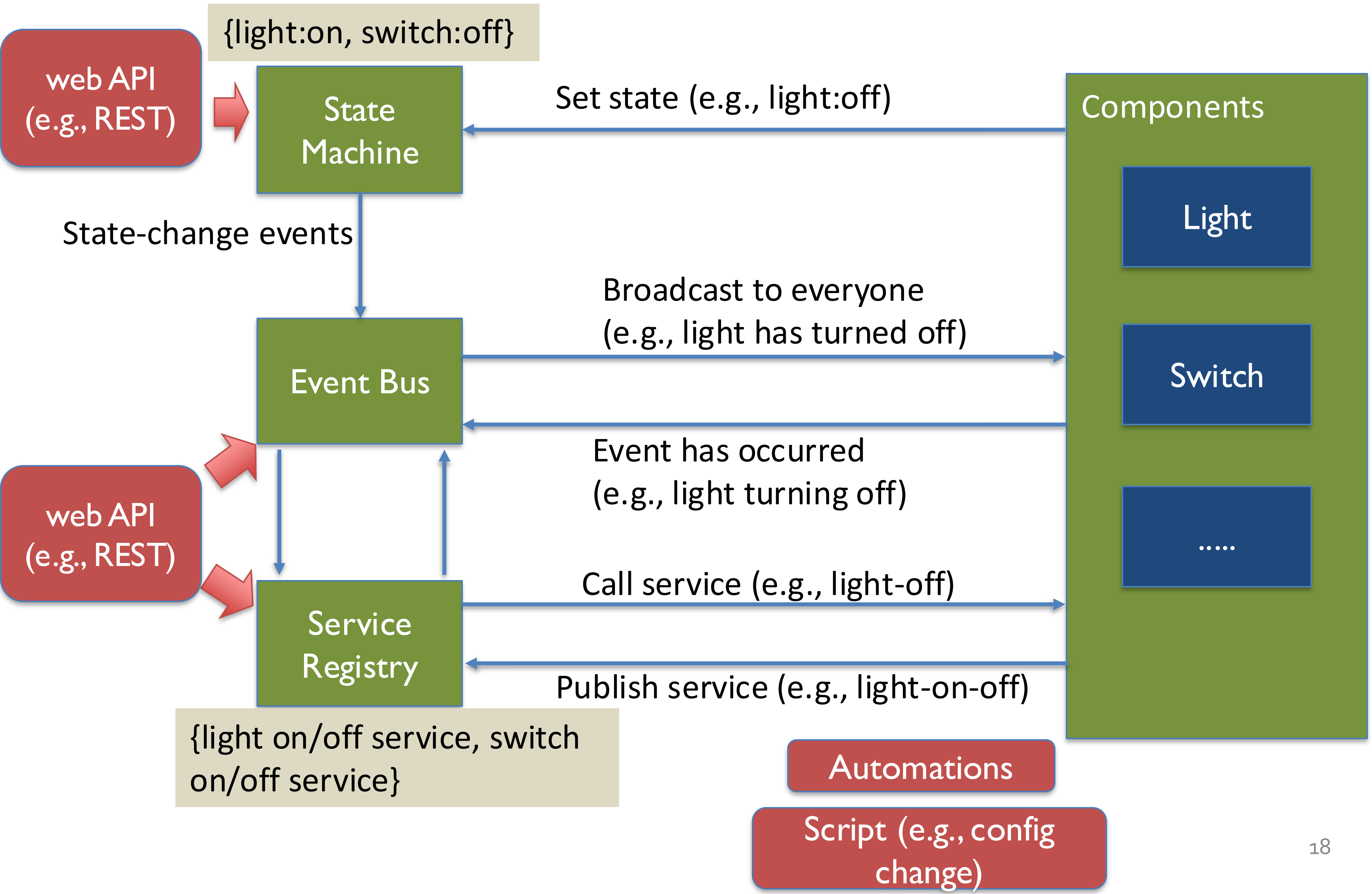
- Allows integrations (e.g., Philips hue, Google home speakers, etc.)
- Allows automations (via UI or via backend)
- Allows scripting to create automations or run scenes
- Can use REST API to interact with the backend!

Architecture


{light:on, switch:off}
State Machine
Set state (e.g., light:off)
Components
Light
State-change events
Broadcast to everyone (e.g., light has turned off)
Switch
Event Bus
Event has occurred (e.g., light turning off)
.....
Call service (e.g., light-off)
Service Registry
Publish service (e.g., light-on-off)
{light on/off service, switch on/off service}

These properties are found in most other smart home platforms as well!

Architecture



{light:on, switch:off}
web API (e.g., REST)
State Machine
Set state (e.g., light:off)
Components
Light
State-change events
Broadcast to everyone (e.g., light has turned off)
Event Bus
Switch
Event has occurred (e.g., light turning off)
web API (e.g., REST)
.....
Call service (e.g., light-off)
Service Registry
Publish service (e.g., light-on-off)
{light on/off service, switch on/off service}
Automations
Script (e.g., config change)

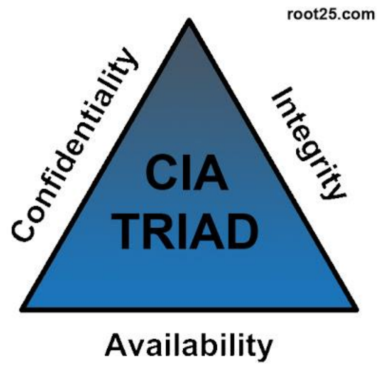
Let us begin..

What is security?

- Garfinkel and Spafford (1991)
  - "A computer is secure if you can depend on it and its software to behave as expected."
- Harrison, Ruzzo, Ullman (1978)
  - "Prevent access by unauthorized users"
- Not really satisfactory – does not truly capture that security speaks to the behavior of others
  - Expected by whom?
  - Under what circumstances?

Security Goals

- *Confidentiality*: Prevention of unauthorized disclosure of information
- *Integrity*: Prevention of unauthorized modification of information
- *Availability*: Prevention of unauthorized withholding of information or resources



Brinkley and Schell, "Concepts and Terminology for Computer Security."

Security Goals (continued)

- *Authenticity*: Related to integrity, but also speaks to the *sender*, as well as *freshness*
- *Secrecy*: Similar to confidentiality, but often used when discussing specific mechanisms, e.g., access control
- *Non-repudiation*: Prevent a party from denying that some action took place e.g., signed through private key, HMACs
- *Privacy*: The ability/right to control access to one's information. There are many definitions. Often conflated with confidentiality/secrecy.

Risk

- *Assets* are valued resources that can be misused
    - Monetary, data (loss or integrity), time, confidence, trust
- *Risk* is the potential for an asset to be misused
    - Many different formulas, e.g., (Risk = likelihood * impact)
    - What does being misused mean?
        - Privacy (personal)
        - Confidentiality (communication)
        - Integrity (personal or communication)
        - Availability (existential or fidelity)

Q: What about a real-world system, say banking?

Threats

- A *threat* is a specific means by which an attacker can put a system at risk
  - An ability/goal of an attacker (e.g., eavesdrop , fraud, access denial)
  - Independent of what can be compromised
- A *threat model* is a collection of threats that deemed important for a particular environment
  - A collection of attacker(s) abilities
  - E.g., A powerful attacker can read and modify all communications and generate messages on a communication channel

Vulnerabilities (attack vectors)

- A *vulnerability* is a systematic artifact that exposes the user, data, or system to a threat
- E.g., buffer-overflow, WEP key leakage
- What is the source of a vulnerability?
    - Bad software (or hardware)
    - Bad design, requirements
    - Bad policy/configuration
    - System Misuse
    - Unintended purpose or environment
        - E.g., student IDs for liquor store

Adversary

- An *adversary* is any entity trying to circumvent the security infrastructure (sometimes called *attacker*)
  - The curious and otherwise generally clueless (e.g., script-kiddies)
  - Casual attackers seeking to understand systems
  - Venal people with an ax to grind
  - Malicious groups of largely sophisticated users (e.g, chaos clubs)
  - Competitors (industrial espionage)
  - Governments (seeking to monitor activities)

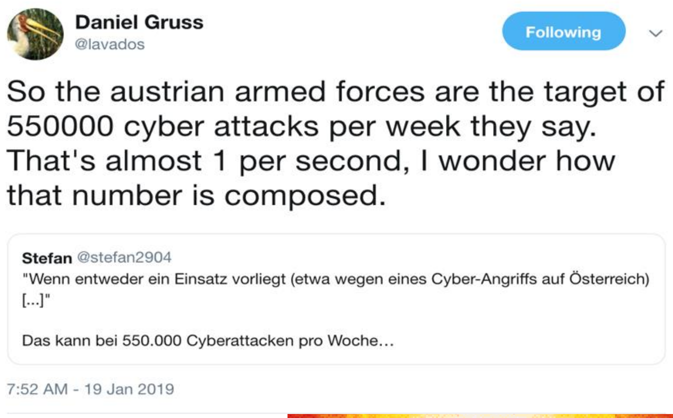
Are users adversaries?

This is known as the <u>insider</u> adversary!

- Have you ever tried to circumvent the security of a system you were authorized to access?
- Have you ever violated a security policy (knowingly or through carelessness)?

Attacks

- An attack occurs when someone attempts to exploit a vulnerability
- Kinds of attacks
  - **Passive** (e.g., eavesdropping)
  - **Active** (e.g., password guessing)
  - **Denial of Service** (DOS)
    - Distributed DOS – using many endpoints



Daniel Gruss
@lavados

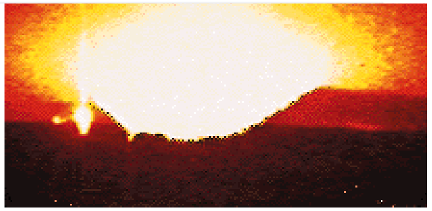
Following

So the austrian armed forces are the target of 550000 cyber attacks per week they say. That's almost 1 per second, I wonder how that number is composed.

Stefan @stefan2904
"Wenn entweder ein Einsatz vorliegt (etwa wegen eines Cyber-Angriffs auf Österreich) [...]"

Das kann bei 550.000 Cyberattacken pro Woche…

7:52 AM - 19 Jan 2019



- A compromise occurs when an attack is successful
  - Typically associated with taking over/altering resources

Participants

- *Participants* are expected system entities
  - Computers, agents, people, enterprises, …
  - Depending on context referred to as: servers, clients, users, entities, hosts, routers, …
  - Security is defined with respect to these entites
    - Implication: every party may have unique view
- A *trusted third party*
  - Trusted by all parties for some set of actions
  - Often used as introducer or arbiter

Q: Example of a trusted third party?

Trust

- *Trust* refers to the degree to which an entity is expected to behave
- What the entity not expected to do?
  - E.g., not expose password
- What the entity is expected to do (obligations)?
  - E.g., obtain permission, refresh
- A *trust model* describes, for a particular environment, who is trusted to do what?
- Note: you make trust decisions every day
  - Q: What are they?
  - Q: Whom do you trust?

Trusted vs. Trustworthy

- *Trusted*: a trusted system or component is one whose failure can break the security policy
- *Trustworthy*: a trusted system or component is one that won't fail

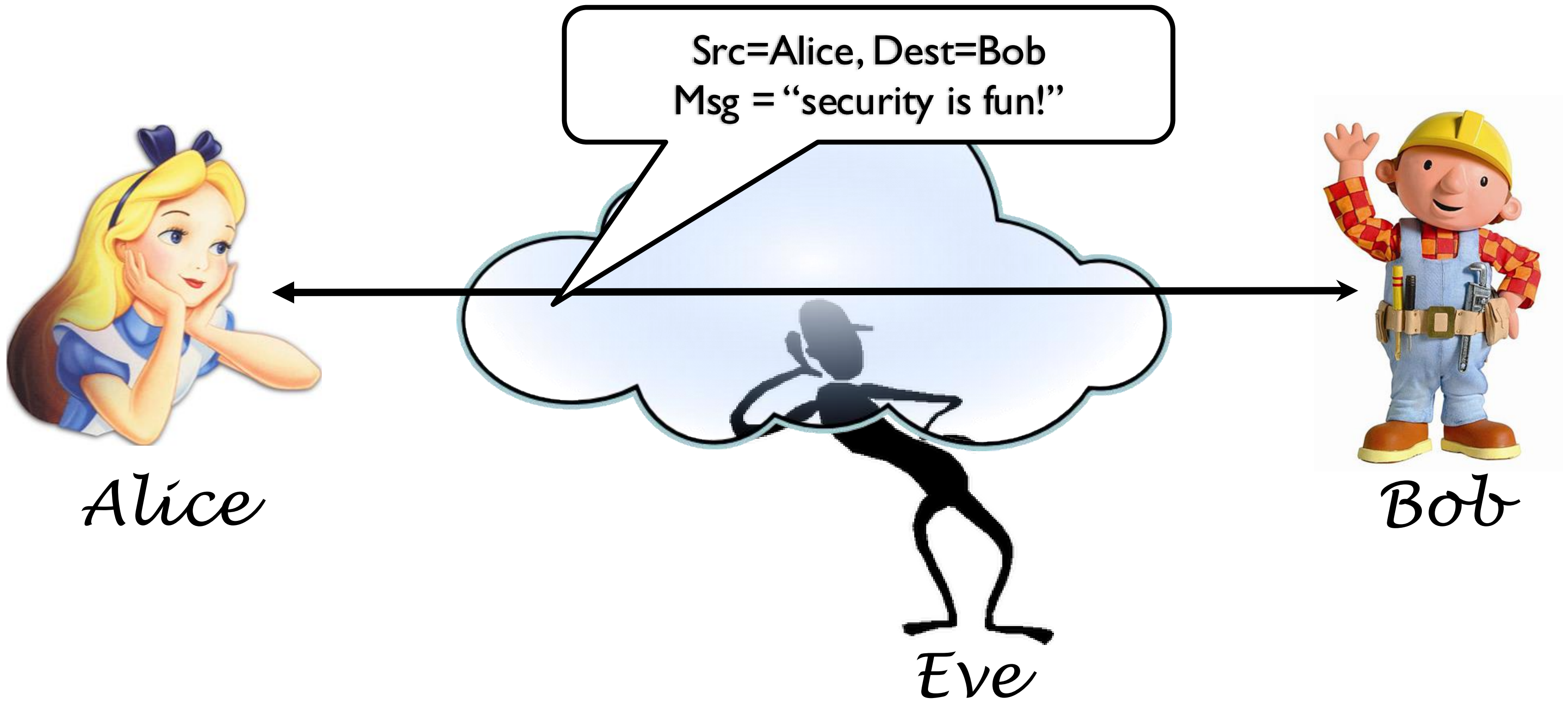
Anderson, "Security Engineering."

Security Model

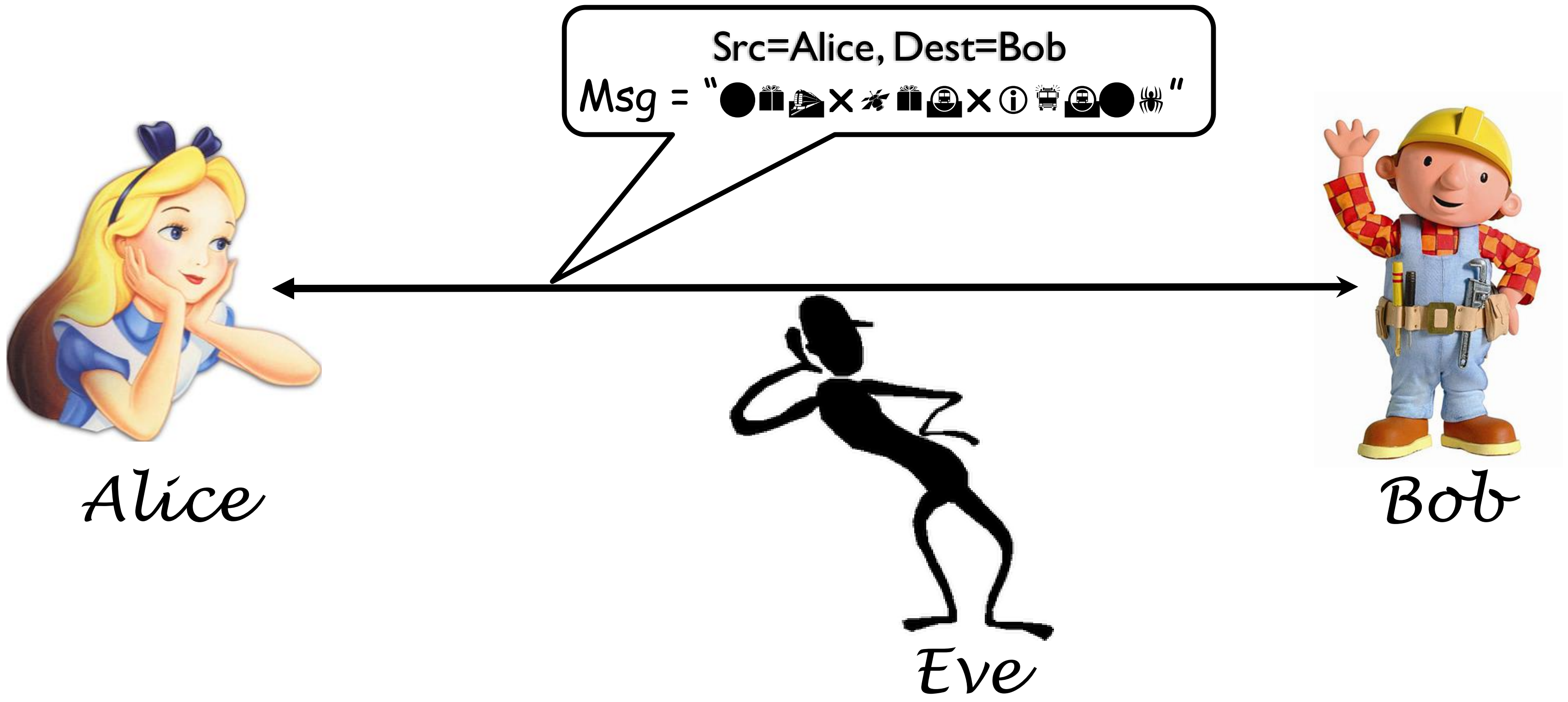
- A *security model* is the combination of a trust and threat models that address the set of perceived risks
    - The "security requirements" used to develop some cogent and comprehensive design
    - Every design must have security model
        - LAN network or global information system
        - Java applet or operating system
- The single biggest mistake seen in use of security is the lack of a coherent security model
    - It is very hard to retrofit security (design time)
- This class is going to talk a lot about security models
    - What are the security concerns (risks)?
    - What are the threats?
    - Who are our adversaries?
    - Who do we trust and to do what?
- Systems must be explicit about these things to be secure.

Let’s look at some potentially desirable properties of a secure network system...

Meet the players.

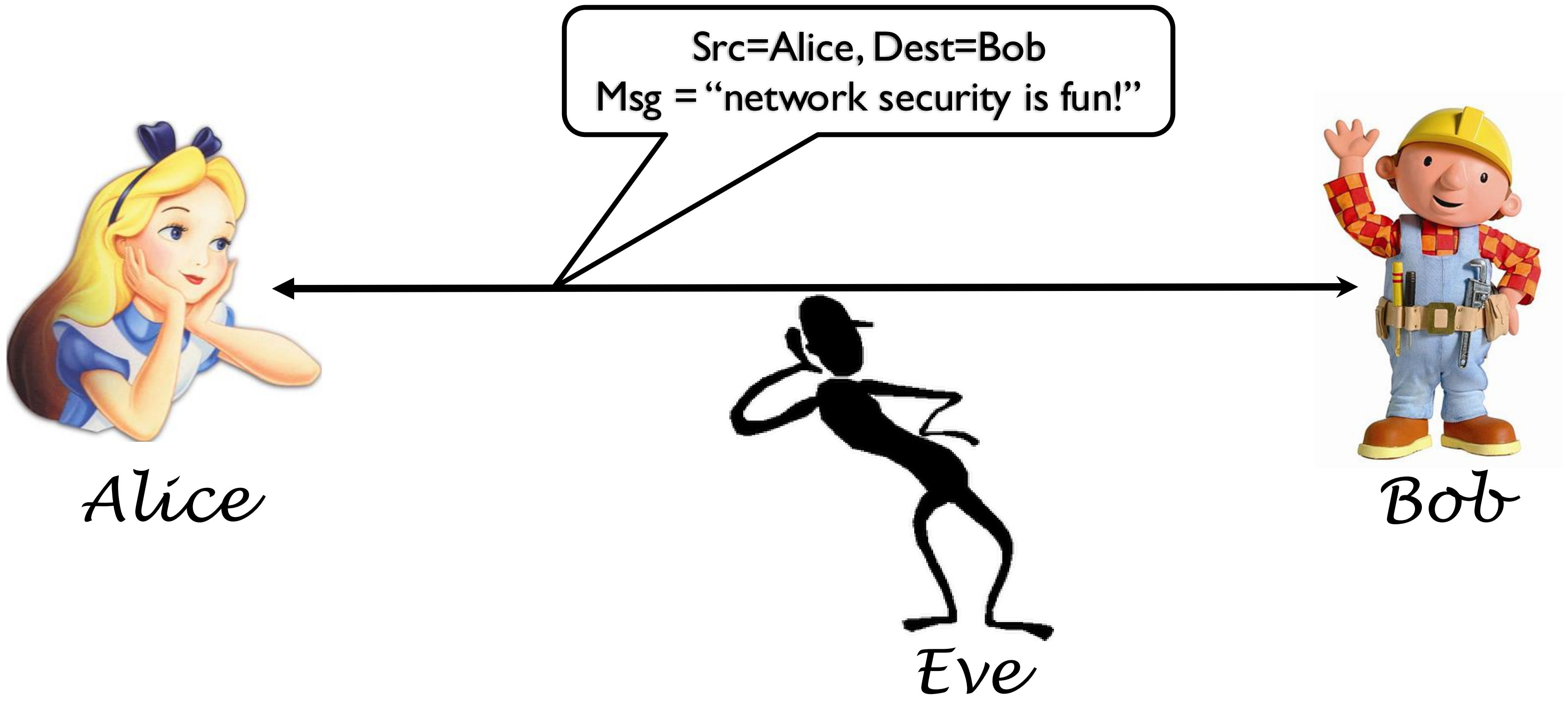


Confidentiality



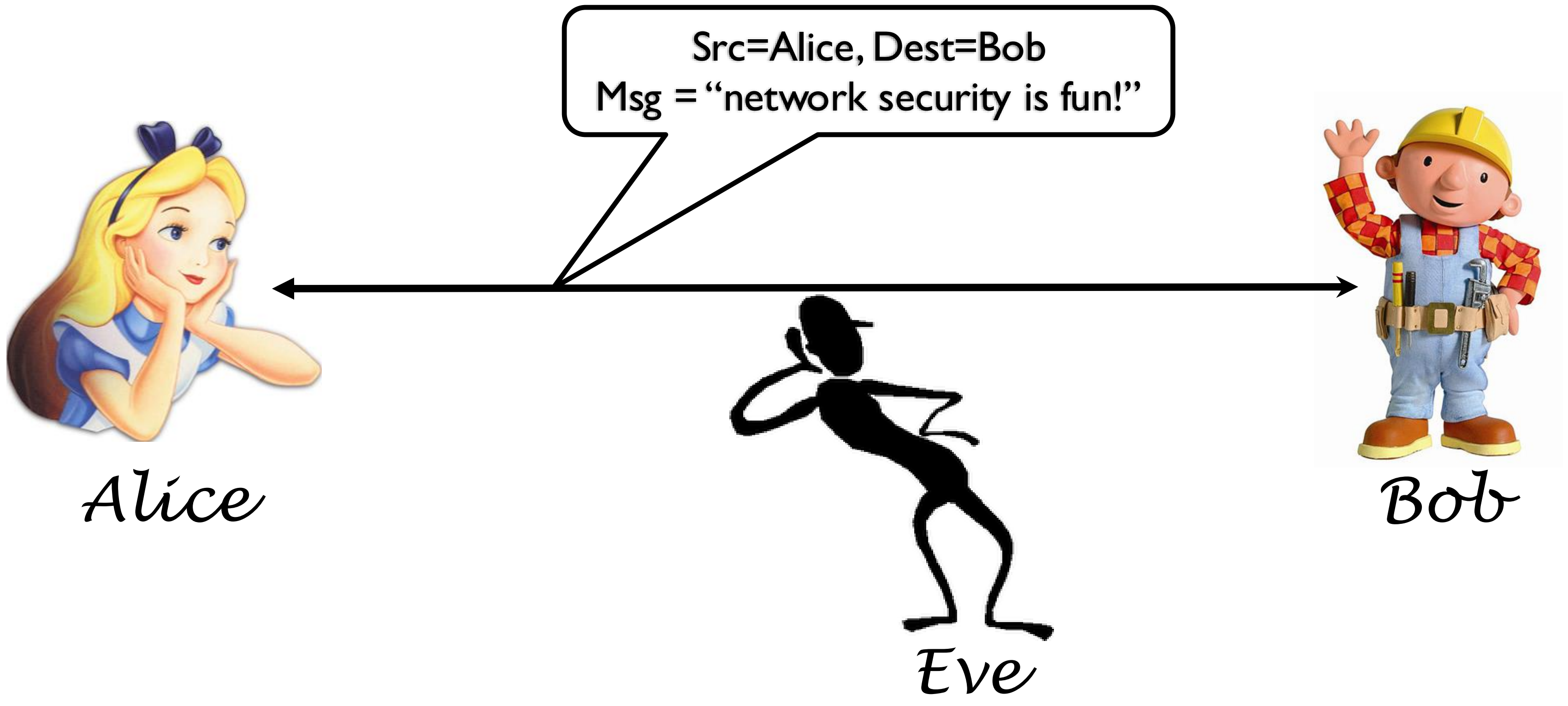
Alice and Bob want to communicate privately, preventing Eve from learning the contents of their communication

Integrity



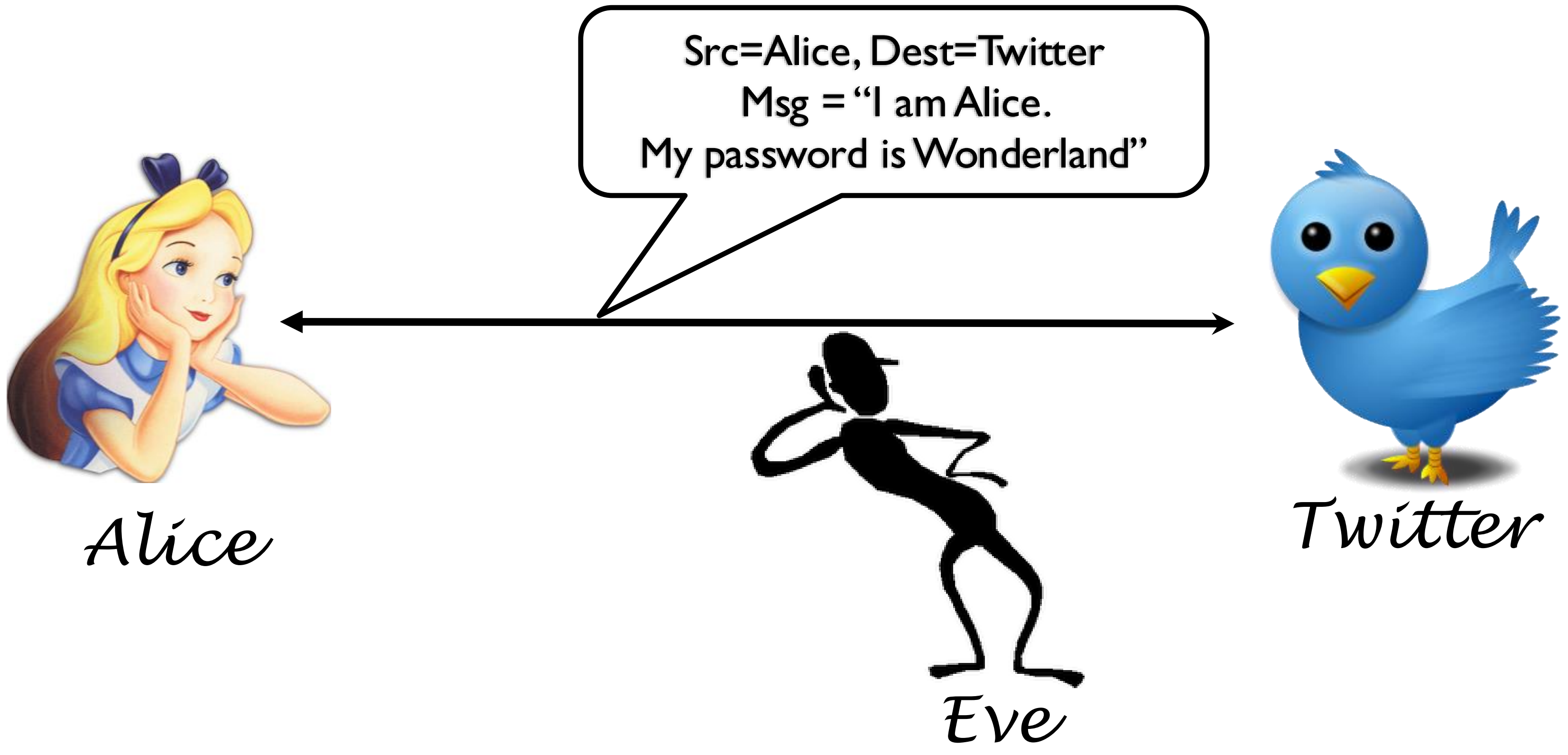
Bob wants to verify that the message hasn't been altered in transit.

Authentication



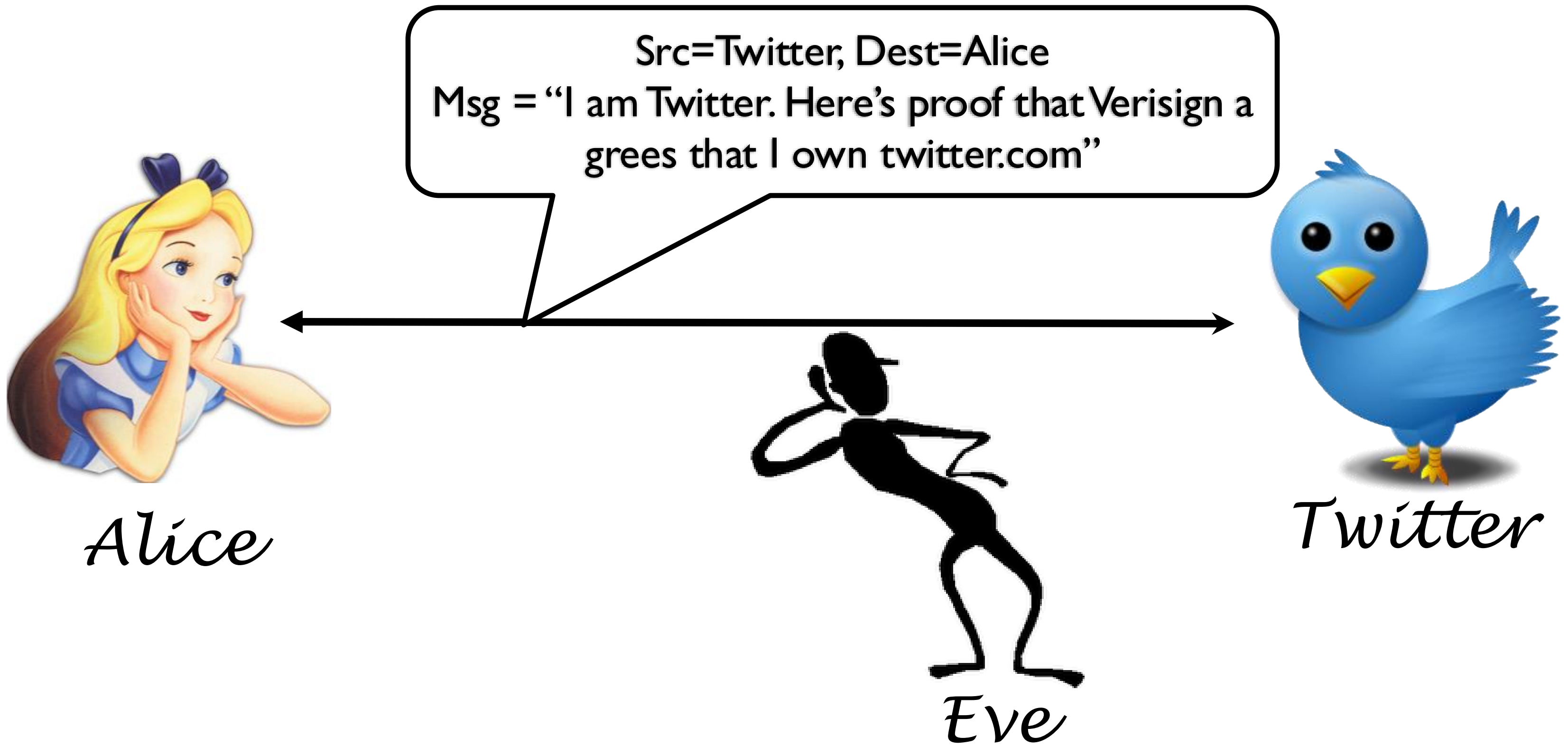
Bob wants to verify that the message is actually from Alice.

Client authentication



Alice wants to prove her identity to the service.

Server authentication


Src=Twitter, Dest=Alice
Msg = "I am Twitter. Here's proof that Verisign a grees that I own twitter.com"
Alice
Twitter
Eve

The service wants to prove its identity to Alice.