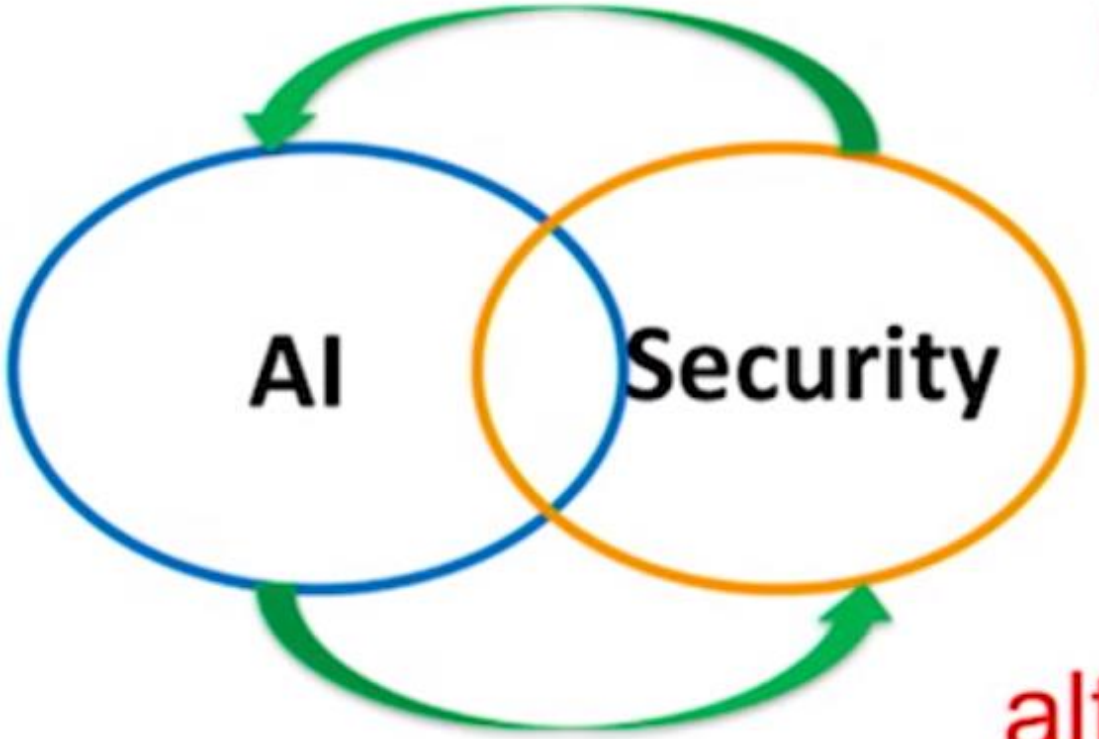
CS 6301.007

Machine Learning in Cyber Security

Wei Yang

Department of Computer Science
University of Texas at Dallas

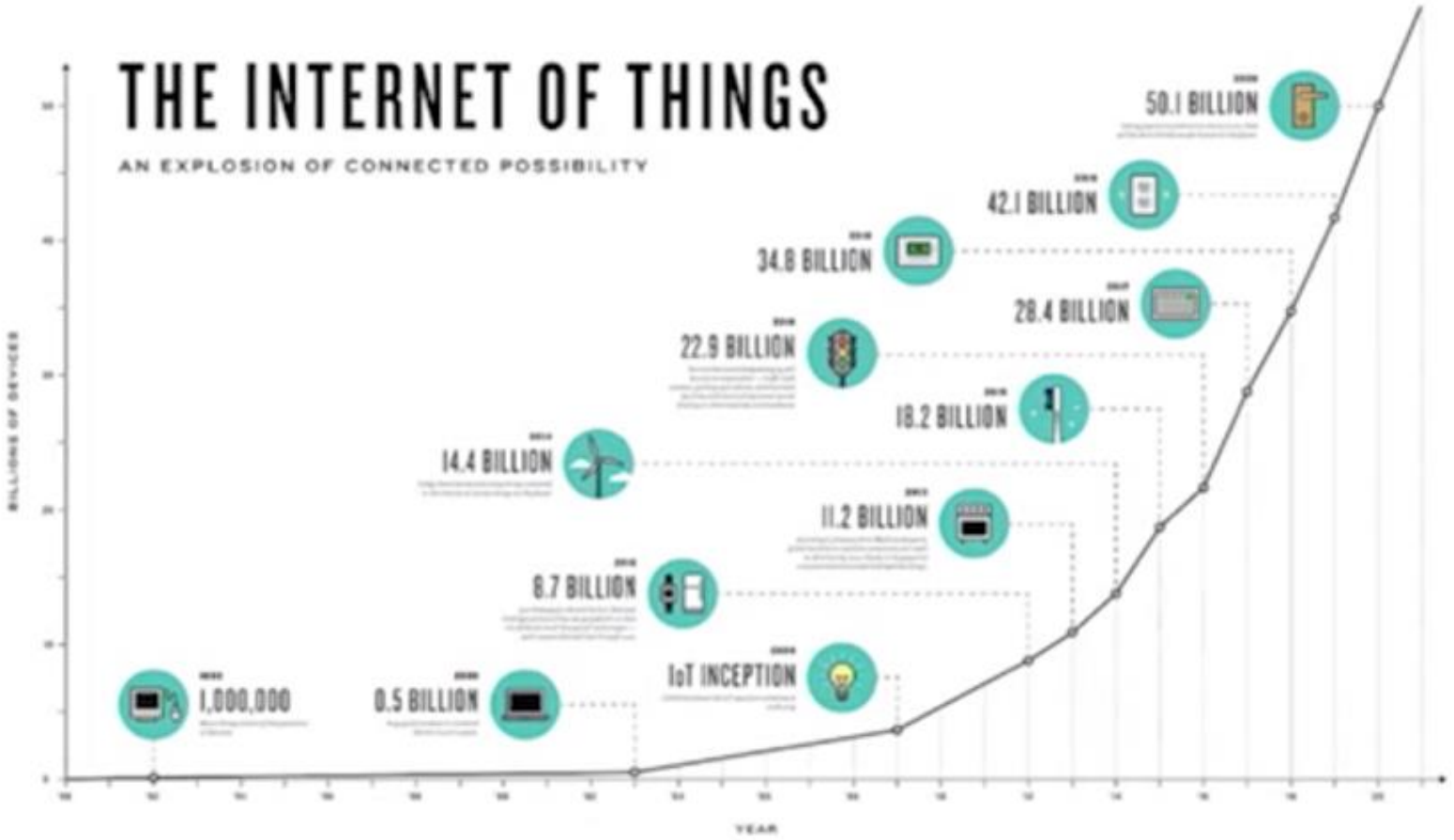


How will (in)security impact the deployment of AI?

How will the rise of AI alter the security landscape?

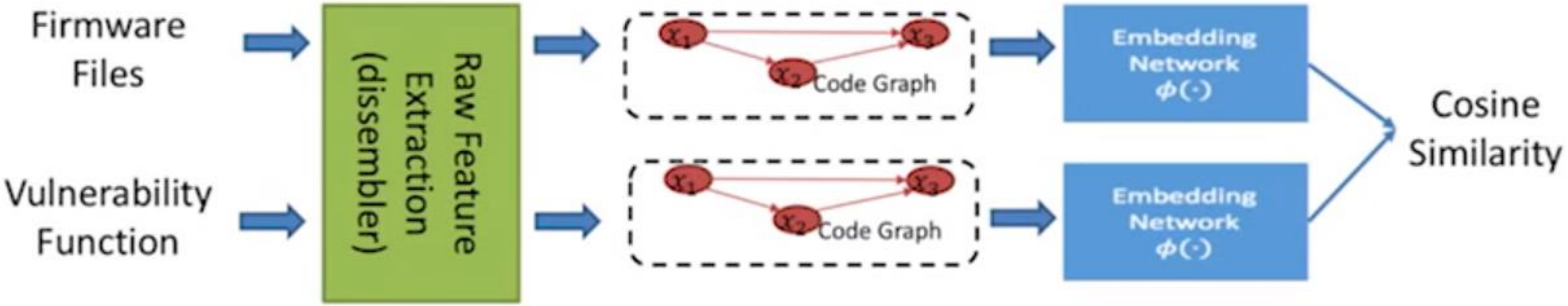
IoT Security

IoT devices are plagued with vulnerabilities from third-party code



Graph Embedding for Binary Code Search

Deep learning for vulnerability detection in IoT Devices



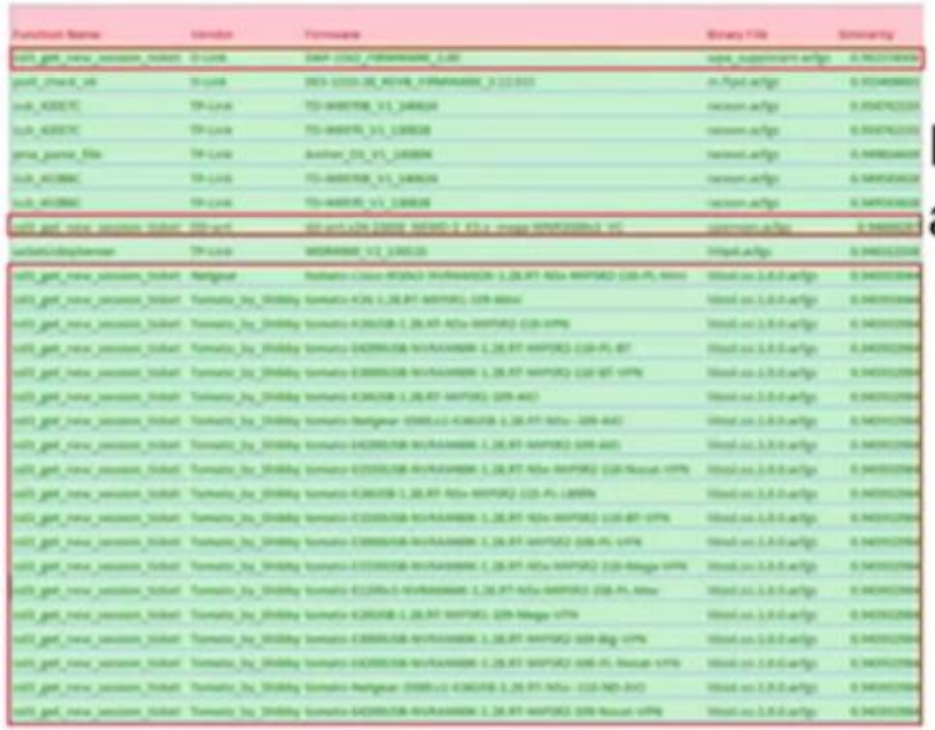
Neural Network-based Graph Embedding for Cross-Platform Binary Code Search
[XLFSSY, ACM Computer and Communication Symposium 2017]

Graph Embedding for Binary Code Search

Deep learning for vulnerability detection in IoT Devices

Training time:
- Previous work: > 1 week
- Our approach: < 30 mins

Serving time (per function):
- Previous work: a few mins
- Our work: a few milliseconds
  - **10,000** times faster



Identified vulnerabilities among top 50:
- Previous work: 10/50
- Our approach: 42/50

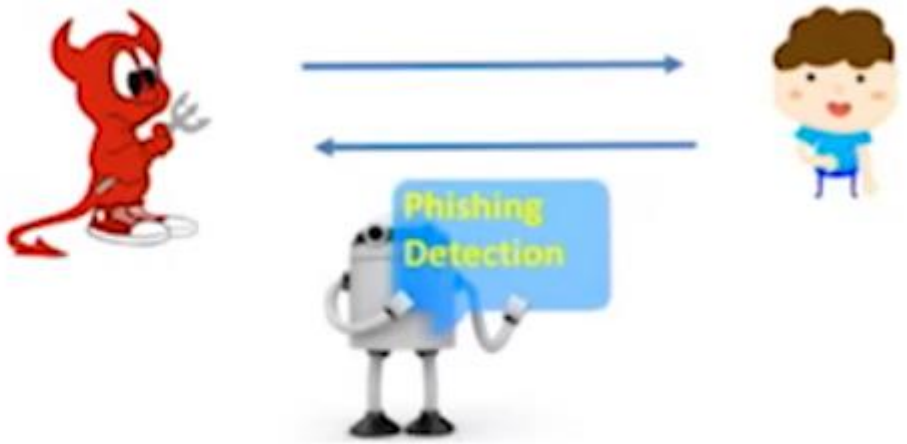
Automatic Agents for security

One fundamental weakness of cyber systems is humans

80+% of penetrations and hacks start with a social engineering attack
70+% of nation state attacks [FBI, 2011/Verizon 2014]

AI Enables Chatbot for Phishing Detection

Chatbot for booking flights, finding restaurants

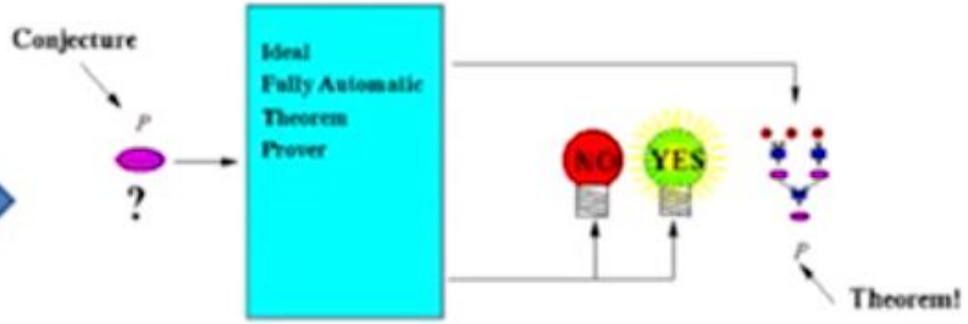


Chatbot for social engineering attack detection & defense

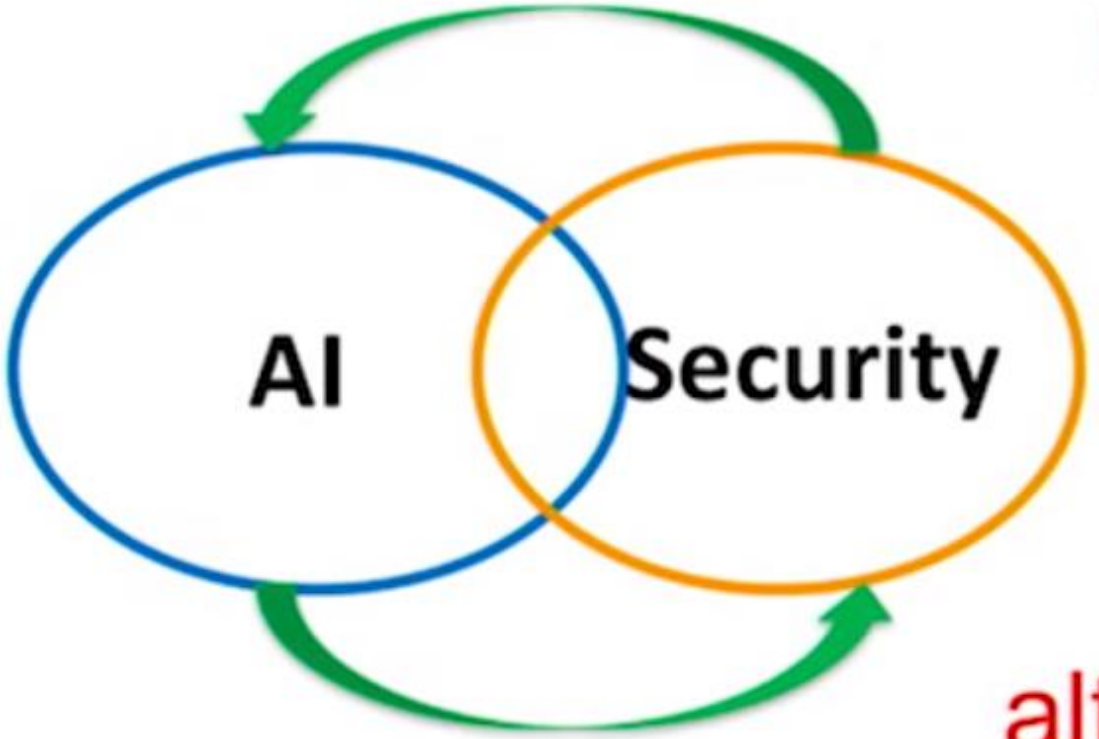
Program Verification

AI Agents to Prove Theorems & Verify Programs

Deep Reinforcement Learning
Agent Learning to Play Go



Automatic Theorem Proving
for Program Verification



How will (in)security impact the deployment of AI?

How will the rise of AI alter the security landscape?

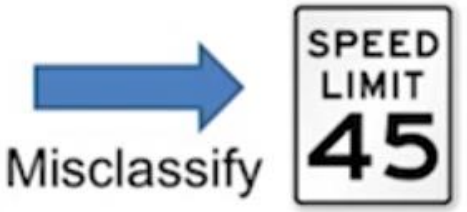
Physical World Attack

Adversarial Examples in Physical World

Adversarial examples in physical world remain effective under different viewing distances, angles, other conditions

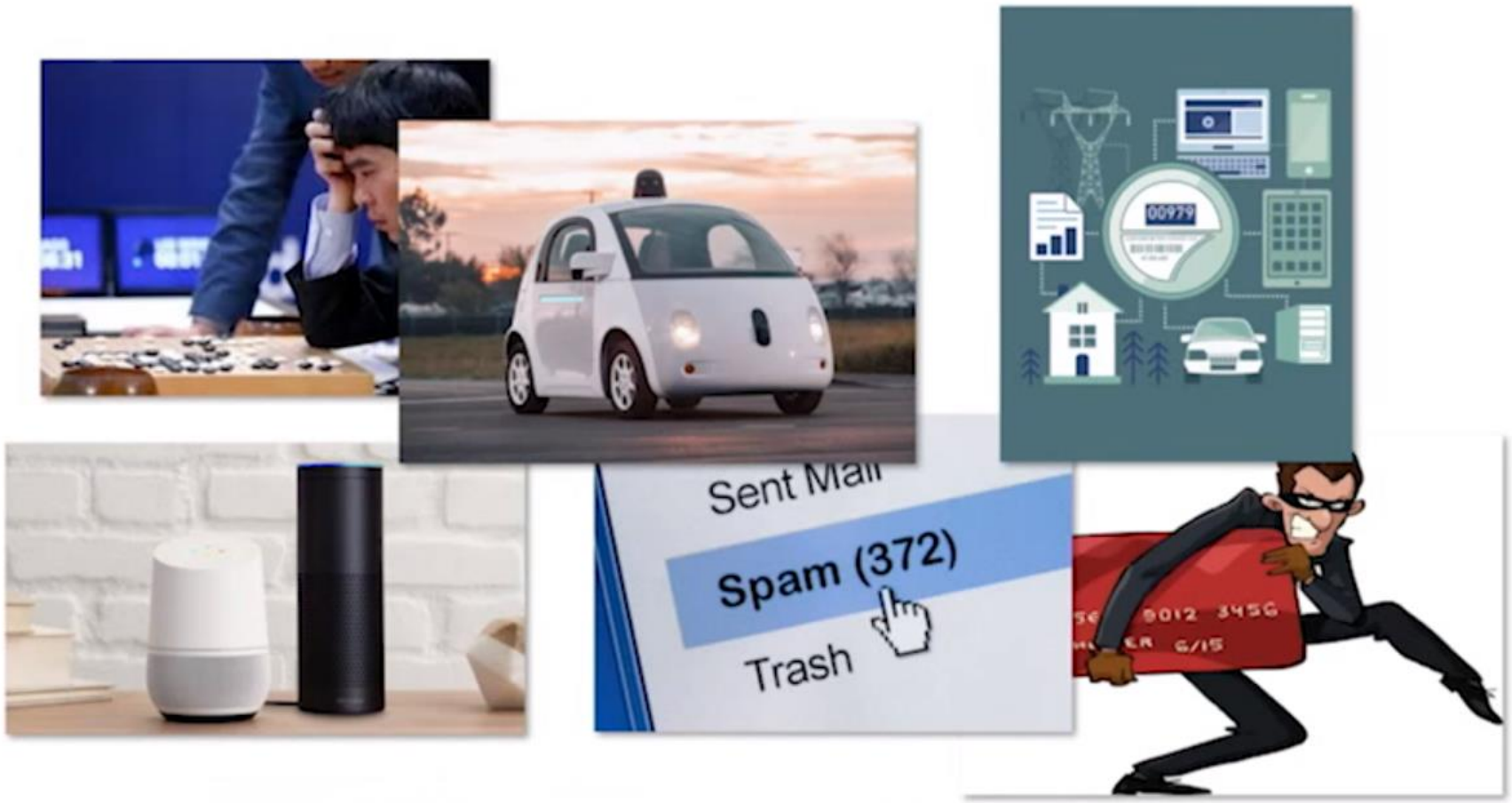
Lab Test Summary (Stationary)

Target Class: **Speed Limit 45**



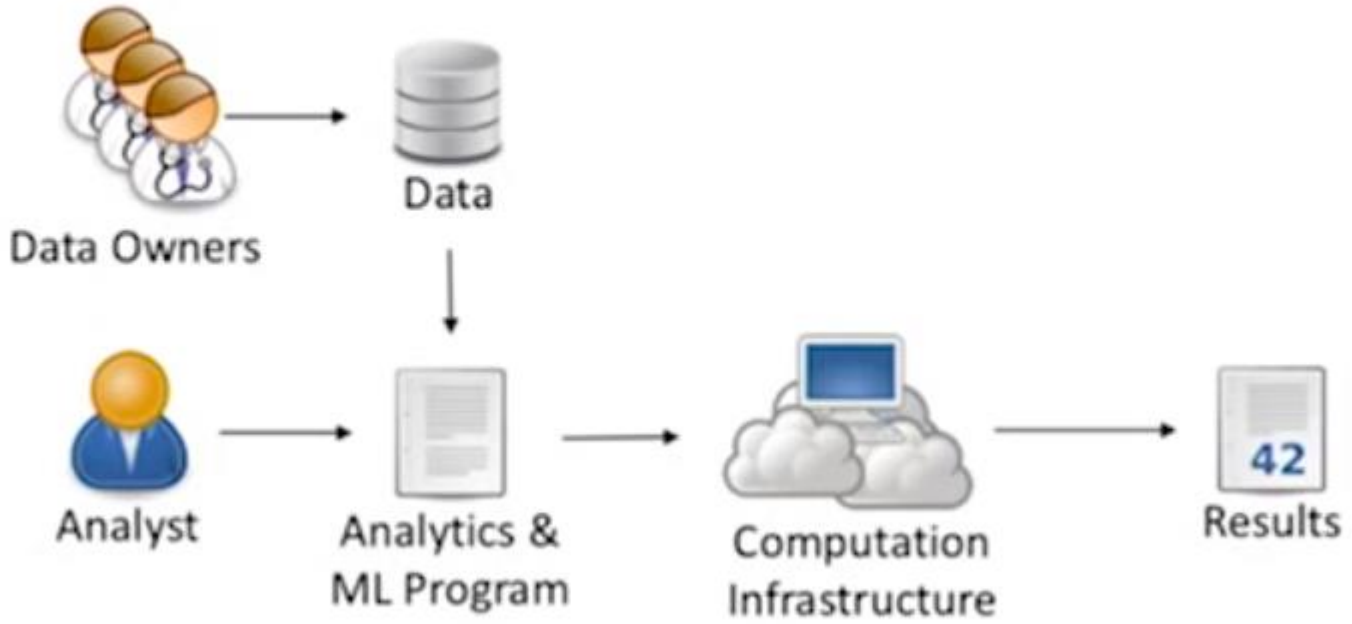
Evtimov, Ivan, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. "Robust Physical-World Attacks on Machine Learning Models." *arXiv preprint arXiv:1707.08945* (2017).

Data is the New Oil

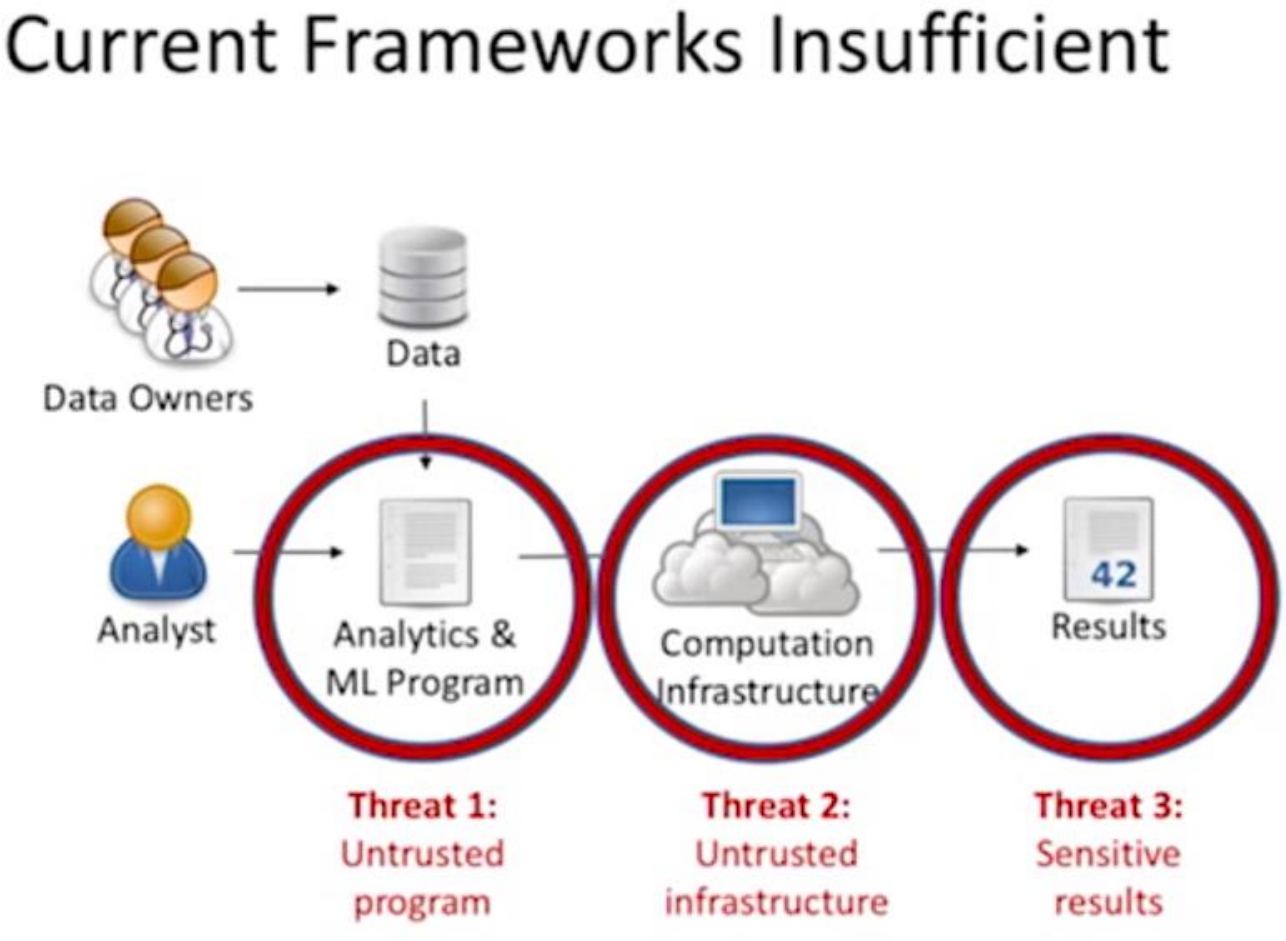


Privacy of ML frameworks

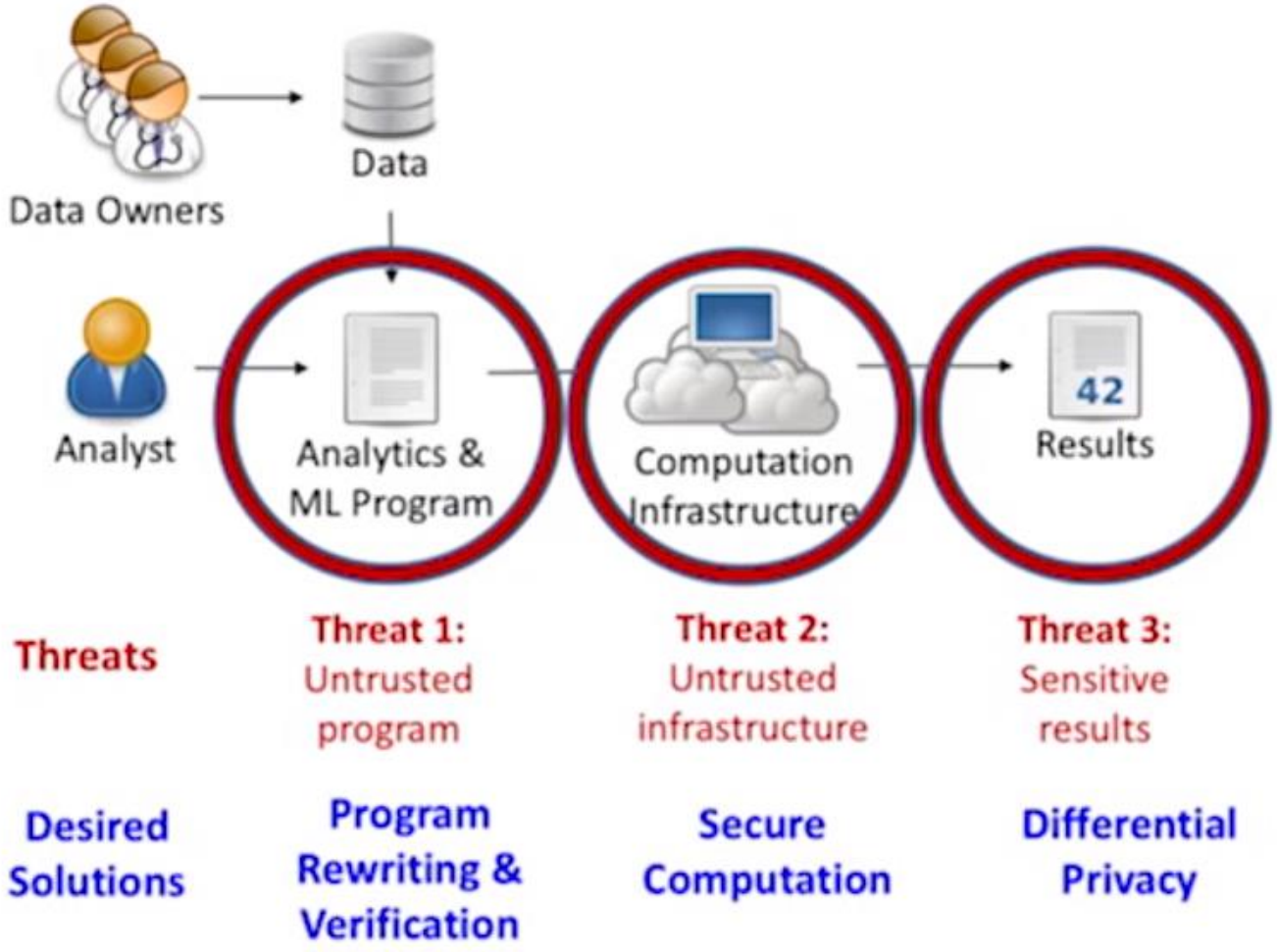
Current Frameworks for Data Analytics & Machine Learning



Privacy of ML frameworks



Desired Solutions for Privacy



What Do Neural Networks Remember?

Can Attackers Extract Secrets (in Training Data) from Learned Models?

N Carlini, C Liu, J Kos, Ú Erlingsson, and D Song.
"The Secret Sharer: Measuring Unintended Neural Network Memorization & Extracting Secrets". 2018.

Case Study: Enron Email Dataset

Train a language model on the Enron Email dataset, which contains actual peoples' credit card and social security numbers.

Partial information is leaked for all secrets; exposure is high.

We are able to extract 3 of the 10 completely from trained models.

User	Secret Type	Exposure	Extracted?
A	CCN	52	✓
B	SSN	13	
C	SSN	16	
	SSN	10	
	SSN	22	
D	SSN	32	✓
F	SSN	13	
G	CCN	36	
	CCN	29	
	CCN	48	✓

Preventing Memorization

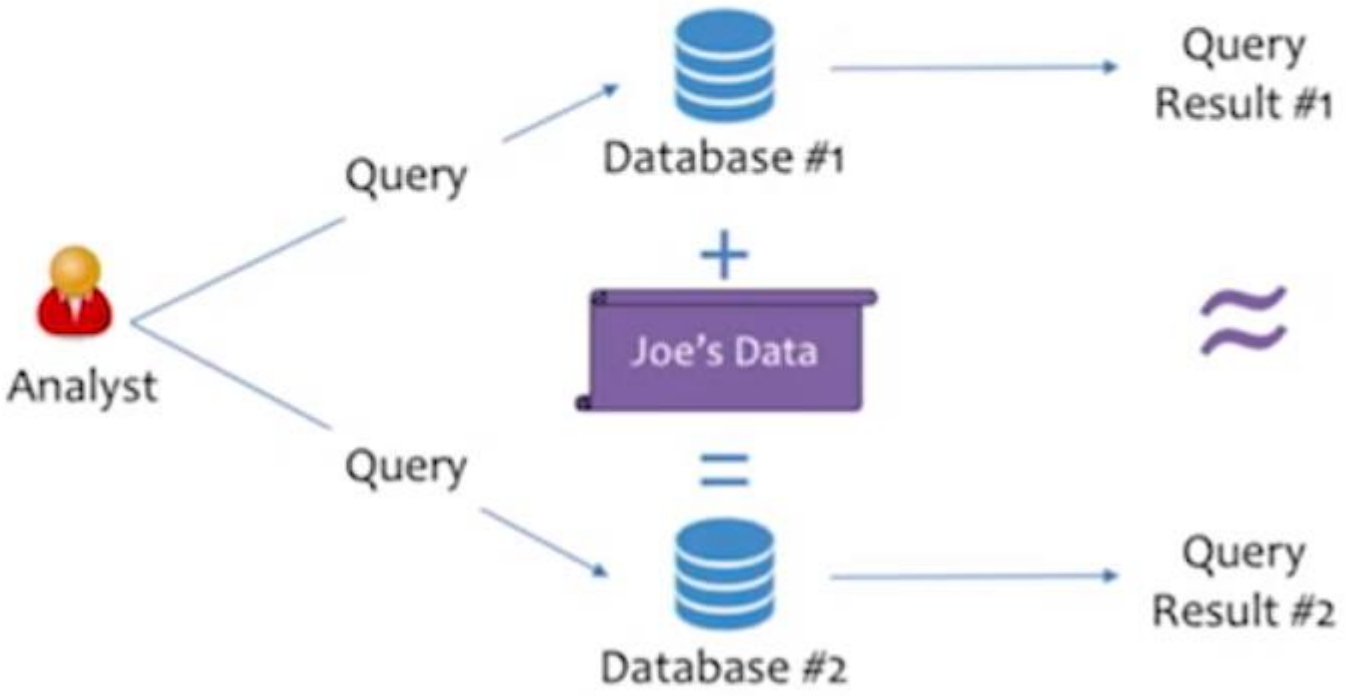
Differential Privacy is a property that a training algorithm satisfies that formally guarantees training data is not revealed

We train a differentially private neural network and empirically observe the exposure is lower, and are unable to extract secrets

	Optimizer	ε	Testing Loss	Estimated Exposure
With DP	RMSProp	0.65	1.69	1.1
	RMSProp	1.21	1.59	2.3
	RMSProp	5.26	1.41	1.8
	RMSProp	89	1.34	2.1
	RMSProp	2×10^8	1.32	3.2
	RMSProp	1×10^9	1.26	2.8
	SGD	∞	2.11	3.6
No DP	SGD	N/A	1.86	9.5
	RMSProp	N/A	1.17	31.0

A formal privacy definition: differential privacy

- Outcome is the same with or without Joe's data



Real-world use of differential privacy

- Previous work on differential privacy is either:
  - Theoretical
  - Targets specialized applications
    - Google: top websites visited
    - Apple: top emojis used
- No real-world deployments of differential privacy for general-purpose analytics

Challenges

Challenges for Practical General-purpose Differential Privacy for SQL Queries

- Usability for non-experts
- Broad support for analytics queries
- Easy integration with existing data environments

No existing system addresses these issues

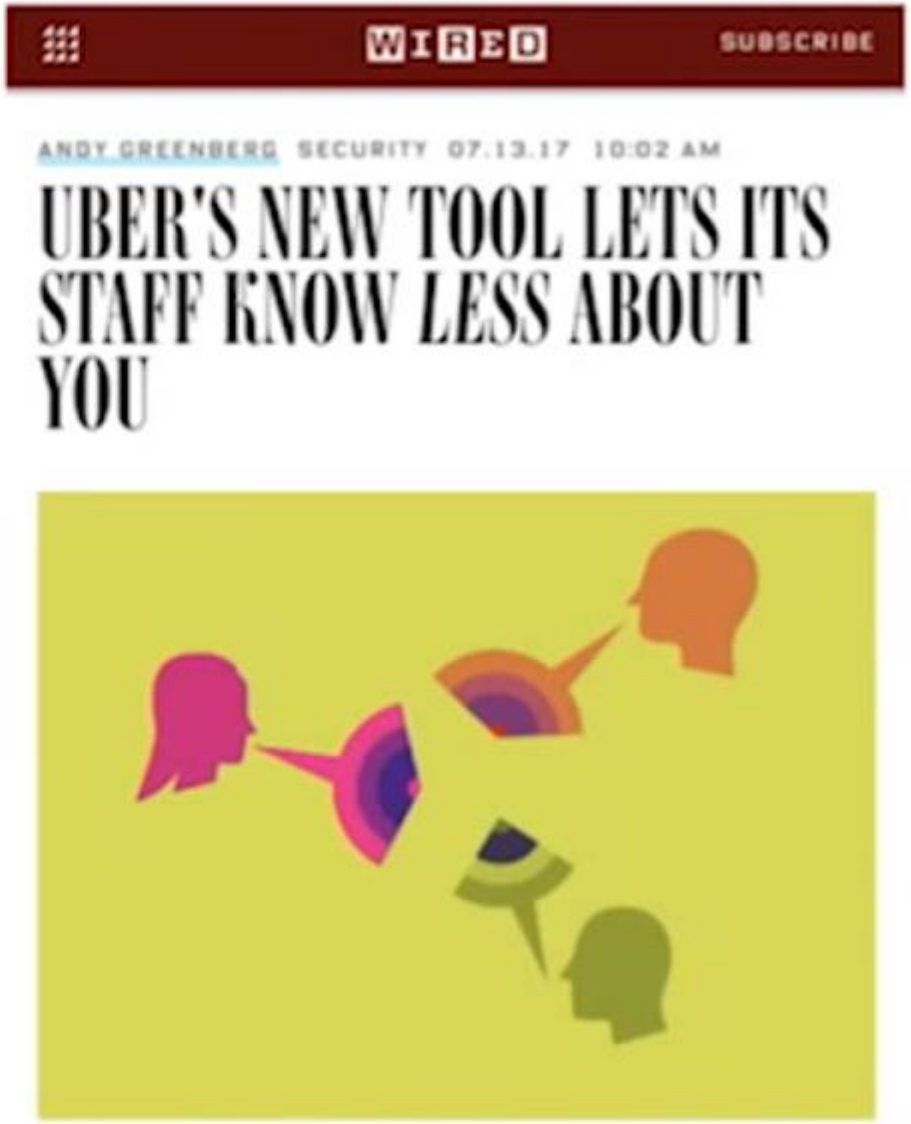
Collaboration with Uber: address practical deployment challenges

Differential Privacy by Query Rewriting

- **Chorus** automatically rewrites input SQL queries into intrinsically private queries
  - Embeds a differential privacy mechanism in the query
  - Does not require any modifications to database engine or data
  - Works with any standard SQL database

Deployment at Uber

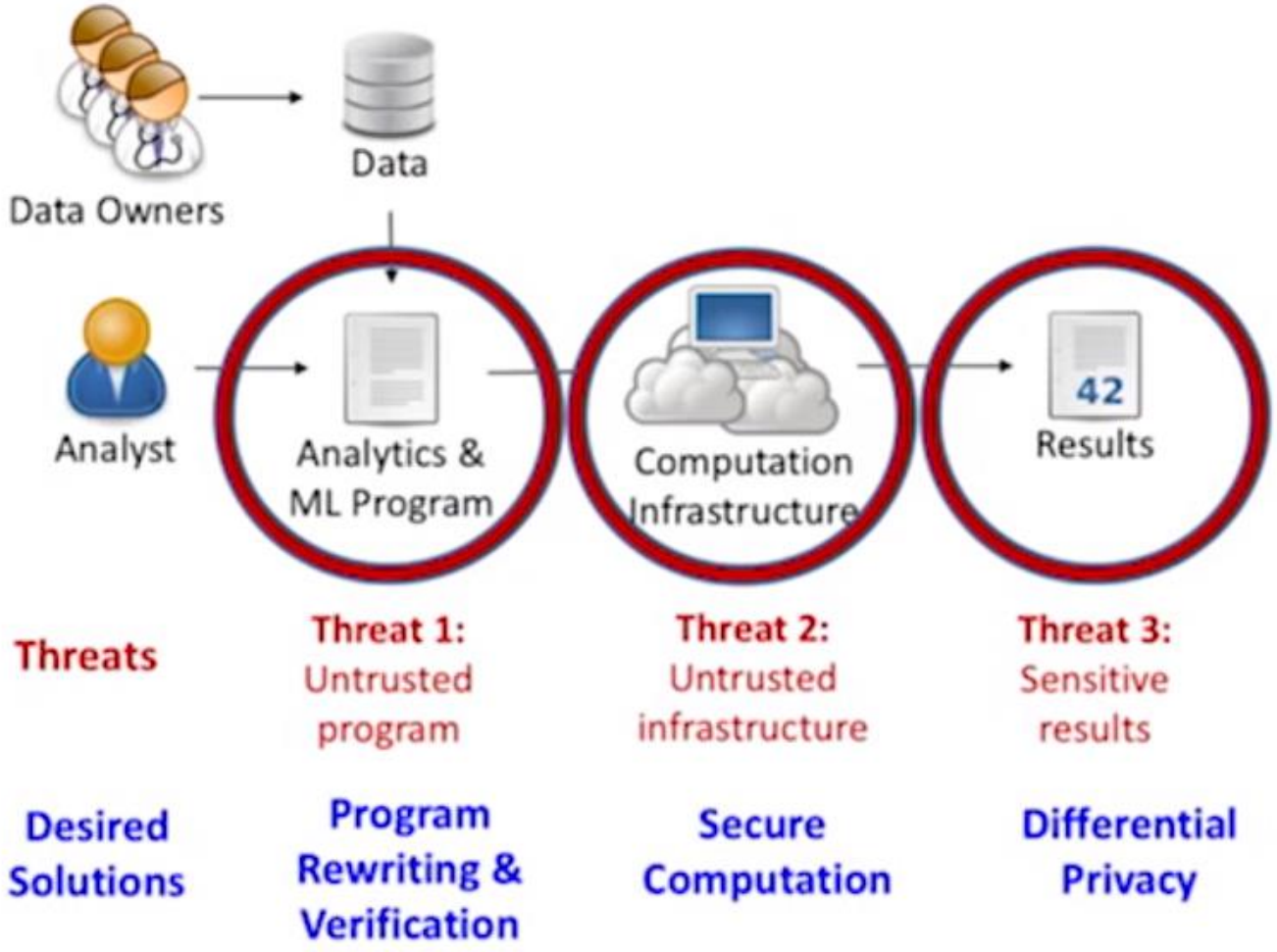
- Ongoing deployment for analytics
  - Differential privacy
  - GDPR
- Plans for public-facing systems
- Open-source release:
  https://github.com/uber/sql-differential-privacy



Optio: Privacy-Preserving Shared Learning Pipelines

- Shared learning pipelines combine analytics & ML tasks
- Optio provides automatic differential privacy guarantees

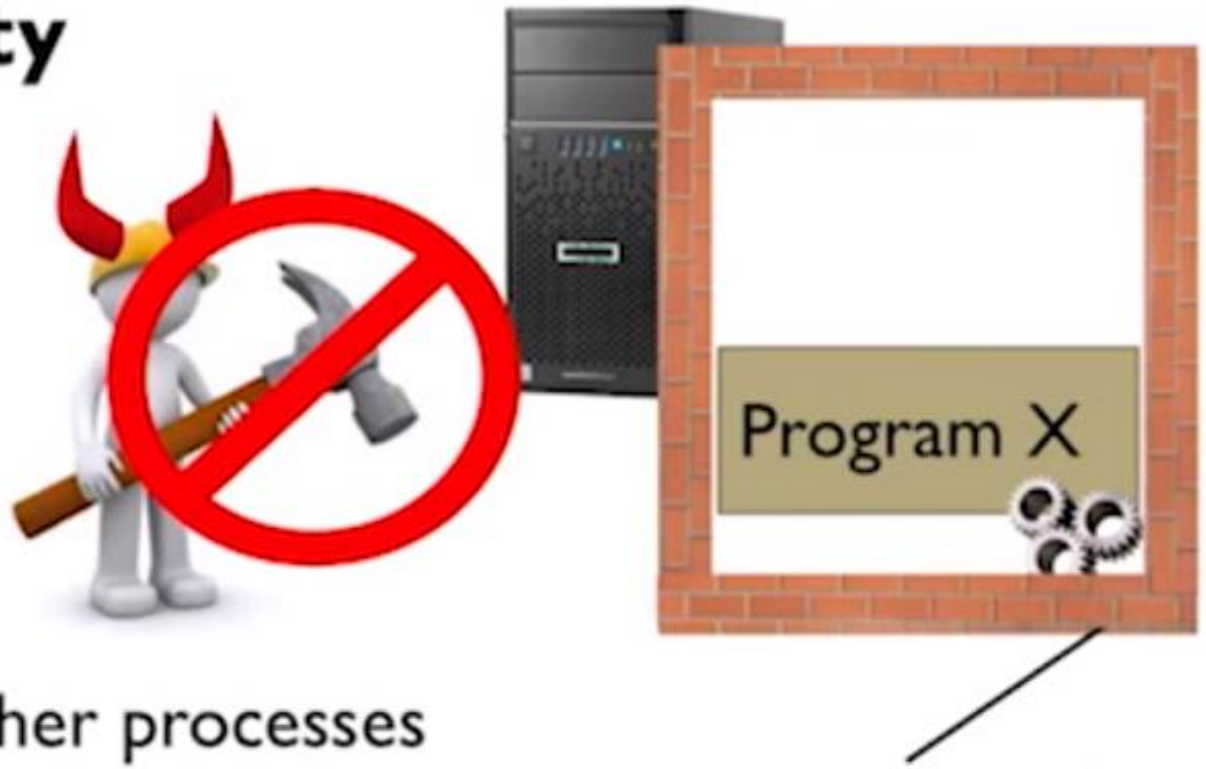
Desired Solutions for Privacy



Trusted Execution Environment (TEE)

Integrity

Confidentiality

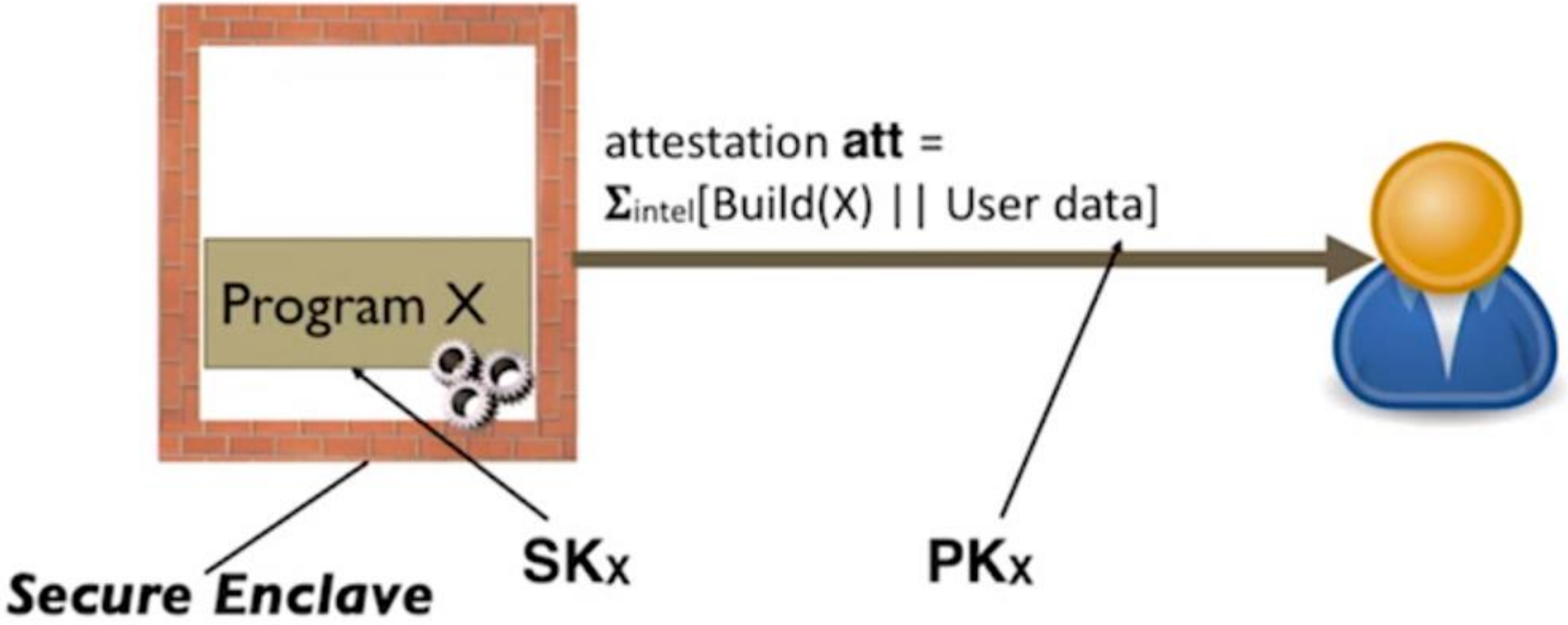


OS and other processes cannot tamper with execution of X.

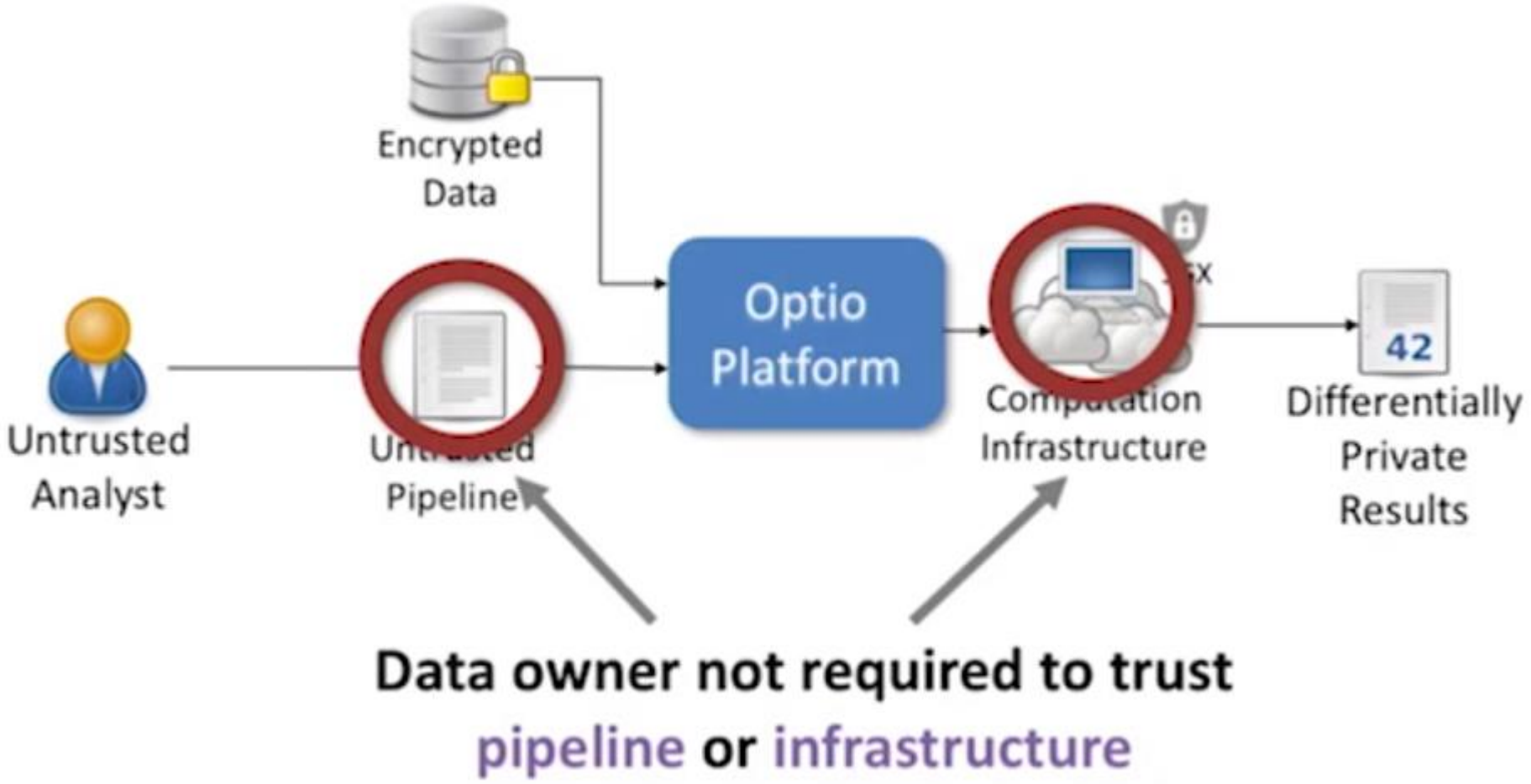
Secure Enclave

OS and other processes cannot learn state of X.*

Remote attestation



A platform for secure, privacy-preserving shared learning



Secure Computation

- Example:
  - Build a word-embedding from everyone's text messages on their phones
- Challenge:
  - Text messages are highly sensitive
  - Computation infrastructure may not be trusted
- Solutions:
  - Crypto-based approach:
    - Non-interactive: Fully-homomorphic encryption (FHE)
    - Interactive: Multi-party computation (MPC)
  - Hardware-based approach:
    - Secure enclave provides isolation & remote attestation

Crypto-based secure computation

- Fully-homomorphic encryption (FHE)
  - Given E(x), f, compute E(f(x))
  - Support general secure computation with strong security
  - High performance overhead: 10^6
  - Example: CryptoNet [Dowlin et al.]
    - Classification of an encrypted image using neural networks
    - On MNIST:
      - 51000 predictions per hour on a single PC
      - 579 seconds latency per image
- Multi-party computation (MPC)
  - Trust model: at most t out of k parties are malicious
  - Require many rounds of communication among different parties

Hardware-based secure computation

- Trusted Execution Environment (e.g., Intel SGX)
  - Secure enclave: isolation & attestation
    - Protect against malicious OS
  - Enable practical secure computation over encrypted data
    - In contrast to fully-homomorphic encryption (FHE) with 10^6 performance overhead
  - Many other security applications

Secure Enclave as a corner stone security primitive

- Strong security capabilities
  - Authenticate “itself (device)”
  - Authenticate “software”
  - Guarantee the integrity and privacy of “execution”
- Platform for building new security applications
  - Couldn’t be built otherwise for the same practical performance
  - Many examples
  - Haven [OSDI’14], VC3 [S&P’15], M2R[USENIX Security’15], Ryoan [OSDI’16], Opaque [NSDI’17]
    - Single node or distributed computation using trusted hardware

Growth in Secure Hardware Deployment

SGX: Software Guard Extensions

- All Intel Core Processors since August 2015 (6th-Generation and later)
- SGX v2 expected to release in relatively near future

AMD Secure Processor

- Built into AMD Accelerated Processing Units (APUs)

SEV: Secure Encrypted Virtualization

- Introduced in EPYC server processor line (2017)
- Provides confidentiality but not integrity

Trusted execution environment

- Hardware-based isolation and integrity for Tegra chipsets
- TLK (Trusted Little Kernel): open-source stack for TEE

GlobalPlatform Trusted Execution Environment

- Already embedded in more than 17 Billion devices

Challenges in secure hardware

- How secure can it be? Under what threat models?

- What would you entrust with secure hardware?
  - Your bitcoin keys
  - Financial data
  - Health data

Challenges in secure hardware

- Can we create trustworthy secure enclave as a corner stone security primitive?
- Widely deployed, enable secure systems on top
- A new secure computation era

Path to trustworthy secure enclave

- Open source design
- Formal verification
- Secure supply-chain management

Importance of open source secure enclave design

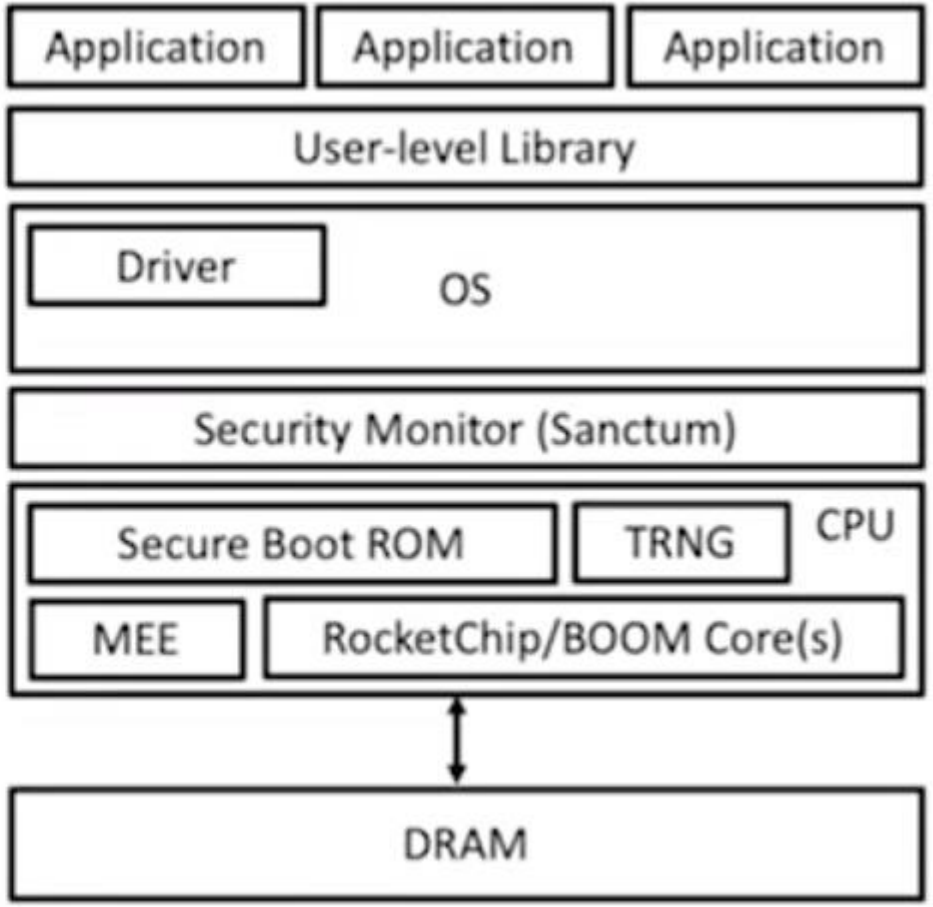
- None of the commercial TEE designs is opened to public
  - Security guarantee relies on trusting a hardware vendor's design
- Open source provides transparency & enables high assurance
- Open source builds a community

RISC-V open-source hardware ecosystem

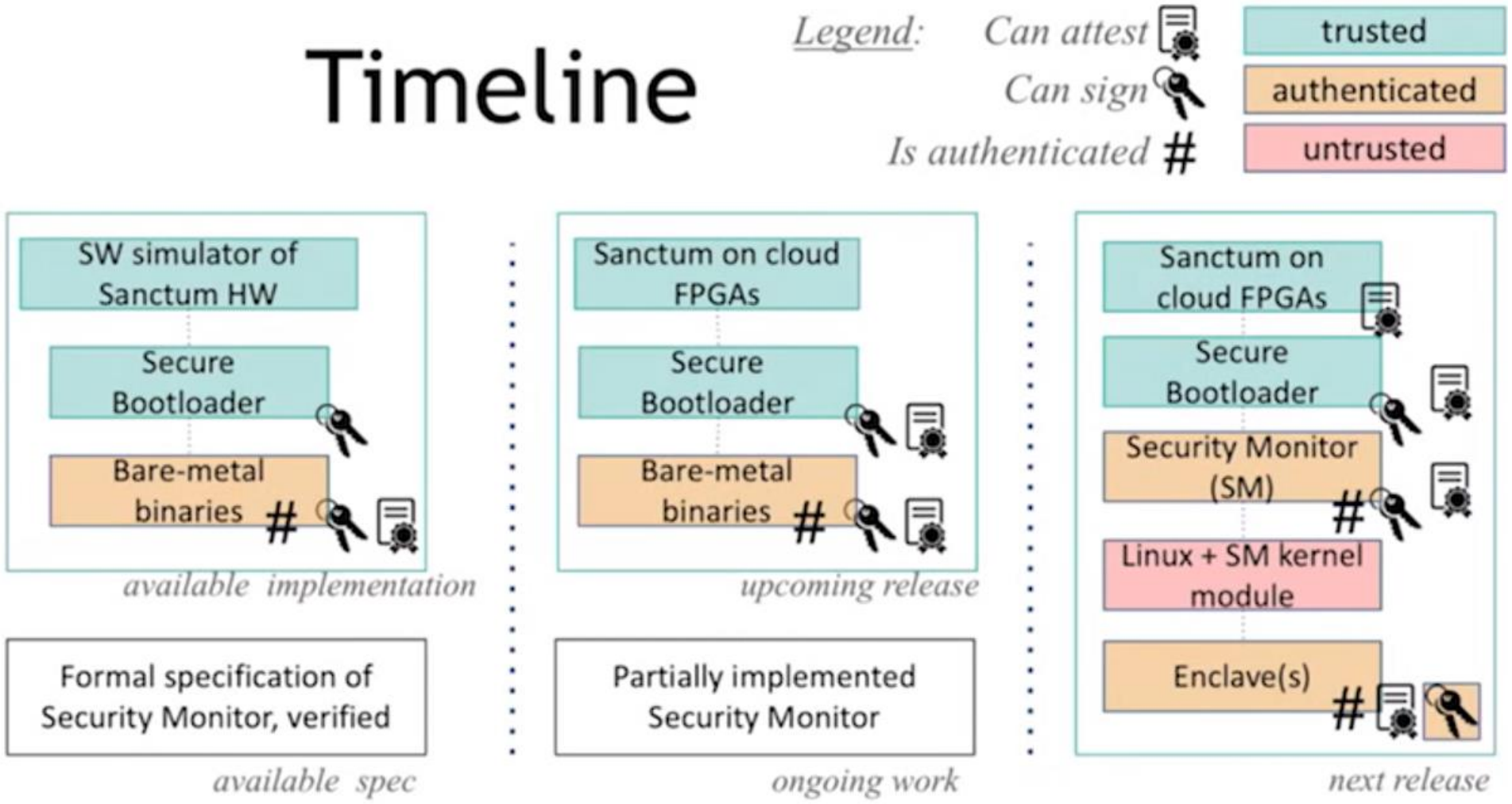
- RISC-V: A high-quality, license-free, royalty-free RISC ISA specification originally from UC Berkeley
- Large companies started to adopt RISC-V for deeply embedded controllers in their SoCs (e.g., NVIDIA, Western Digital)
- India government, US DARPA, and Israel have adopted RISC-V
- Many startups choosing RISC-V for new products
- Becoming standard ISA for academic research

Keystone-enclave: Open source secure enclave

- Full-stack open-source hardware enclave implementation for RISC-V processors
- CPU
  - RocketChip/BOOM: Berkeley-built Open-Source Cores
  - TRNG
  - Memory Encryption Engine (MEE)
- Hardware Enclave
  - MIT Sanctum: Software-based Hardware Enclave
- OS, Library, and Applications



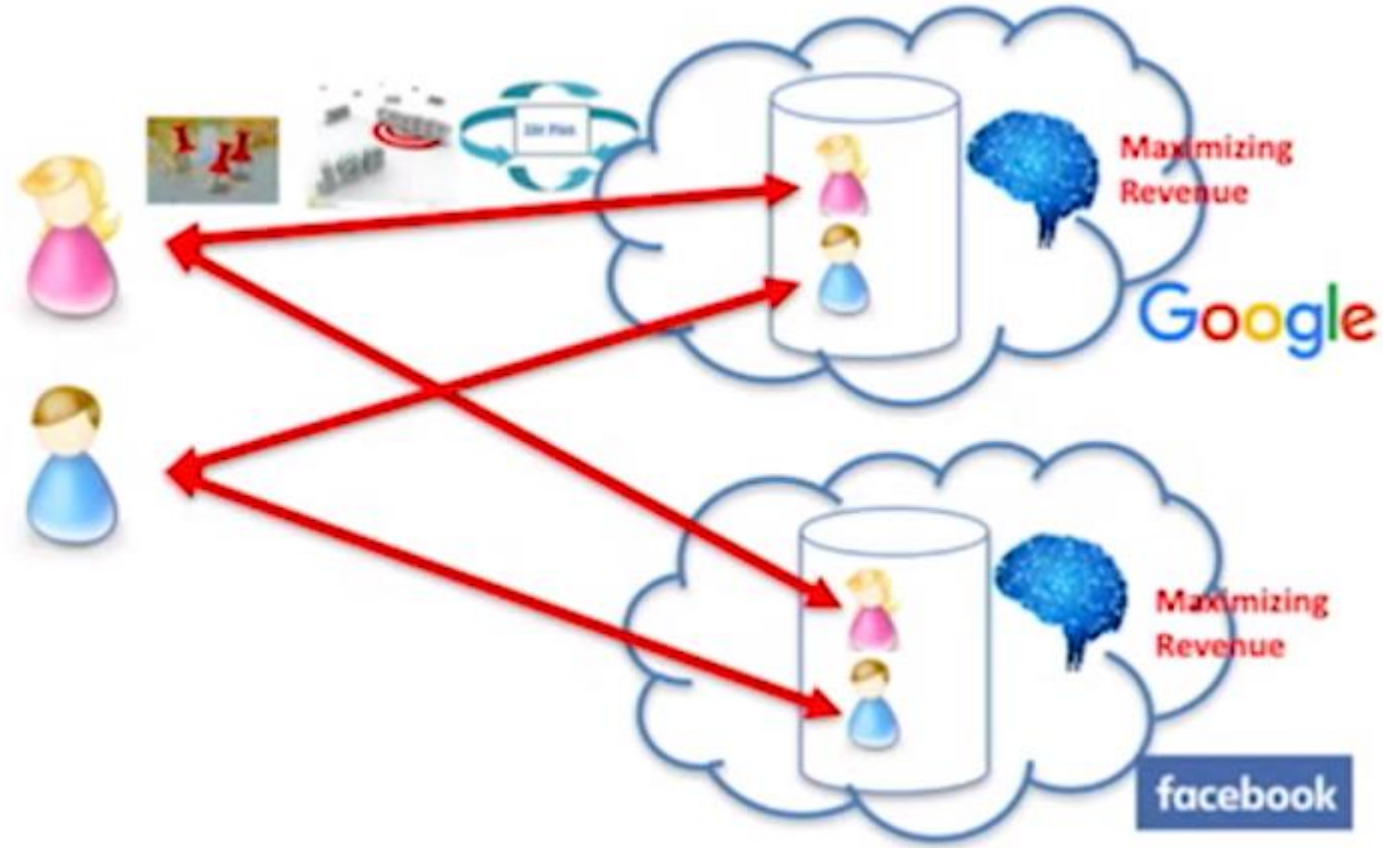
Timeline


Legend:
Can attest
Can sign
Is authenticated #
trusted
authenticated
untrusted
SW simulator of Sanctum HW
Secure Bootloader
Bare-metal binaries #
available implementation
Formal specification of Security Monitor, verified
available spec
Sanctum on cloud FPGAs
Secure Bootloader
Bare-metal binaries #
upcoming release
Partially implemented Security Monitor
ongoing work
Sanctum on cloud FPGAs
Secure Bootloader
Security Monitor (SM) #
Linux + SM kernel module
Enclave(s) #
next release

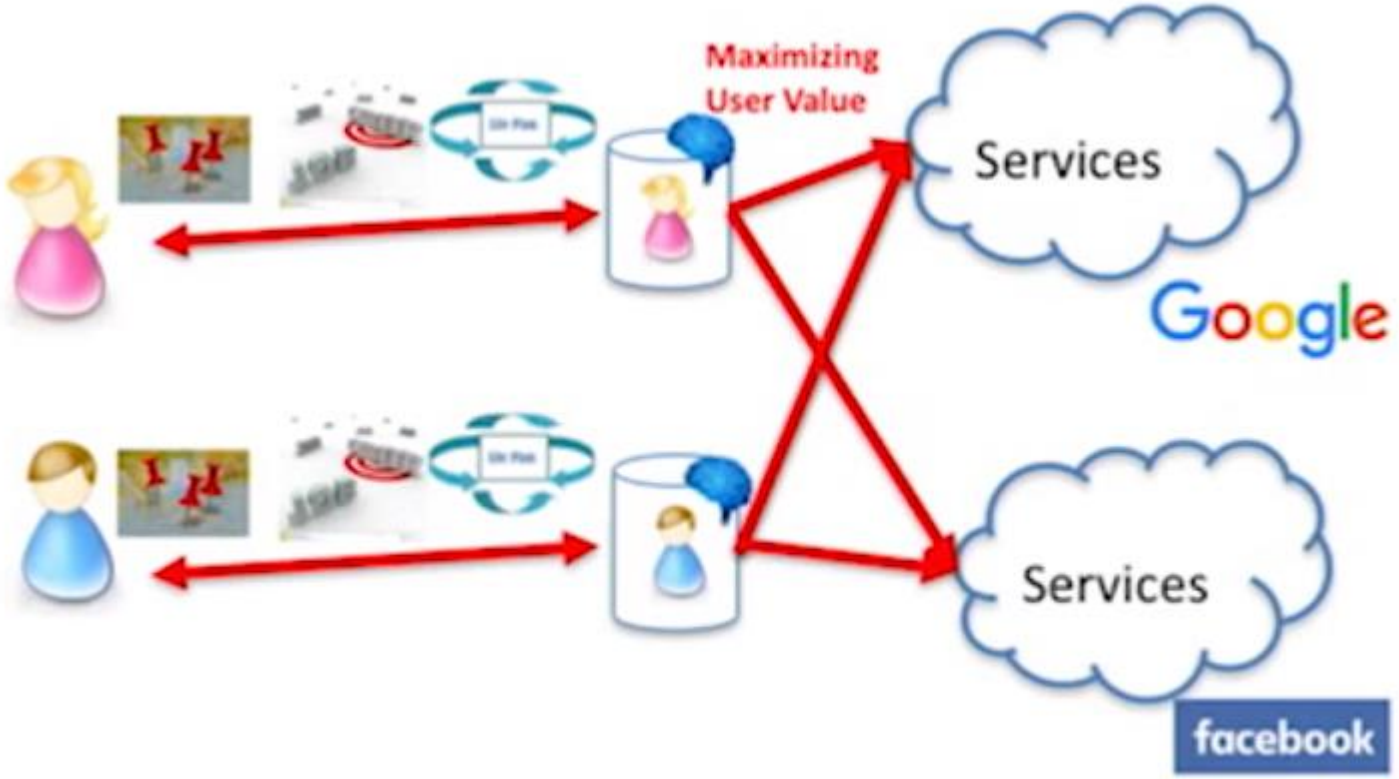
Whoever Controls & Leads in AI Will Rule the World

--Nation State Leaders

Who will be running our lives?



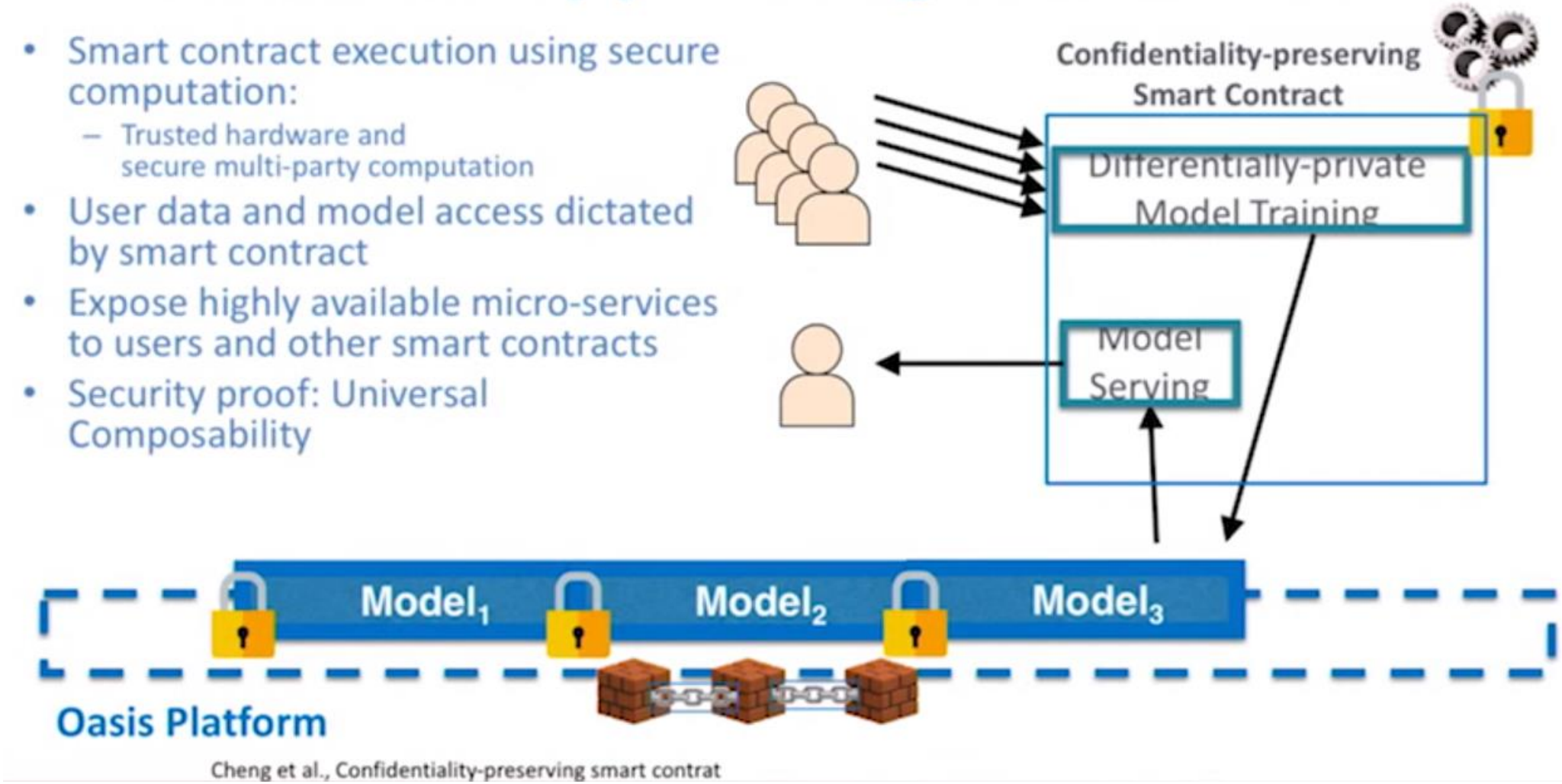
Intelligent agent under user control



Next Generation Blockchain for Democratizing AI

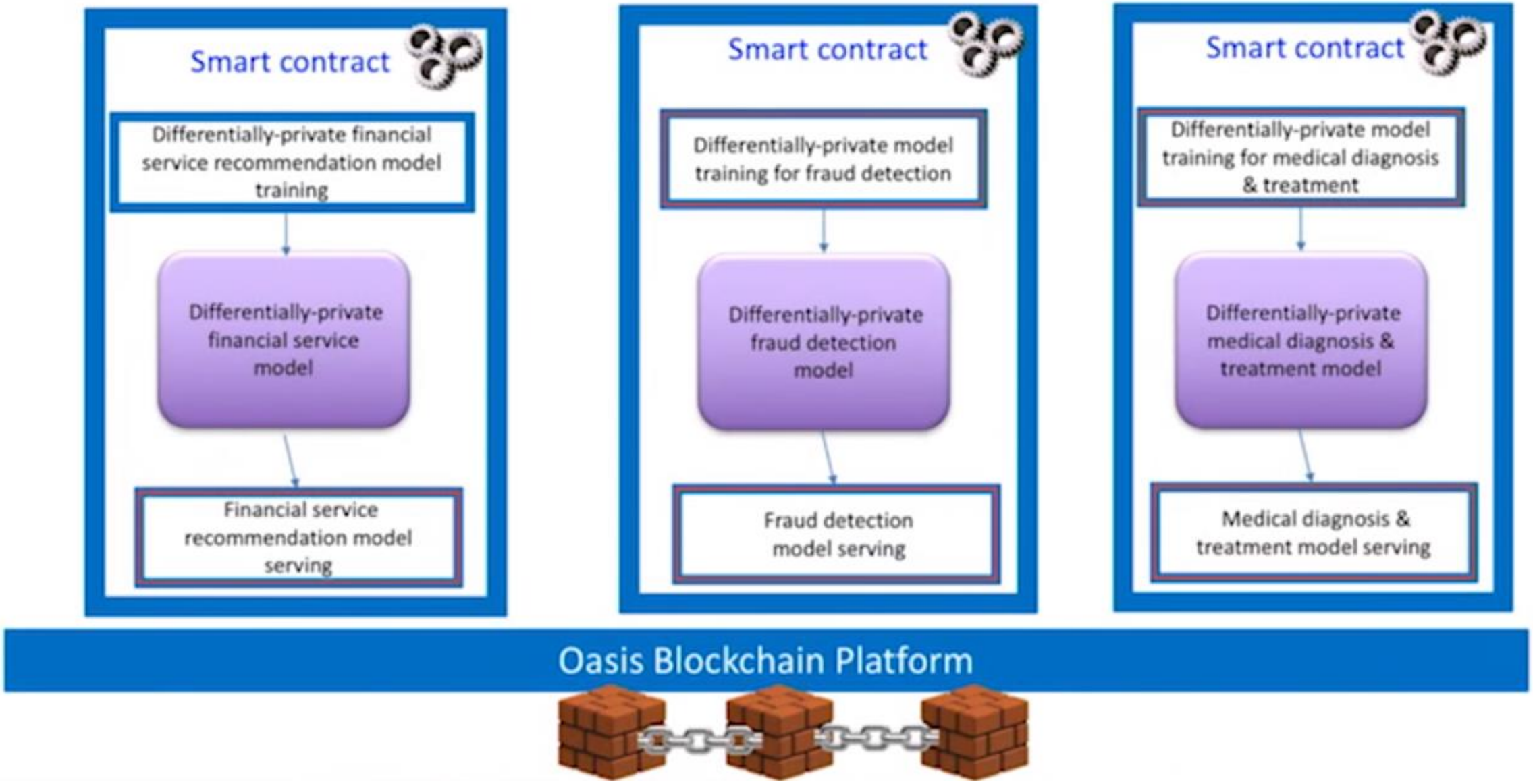
Confidentiality-preserving smart contract

- Smart contract execution using secure computation:
  - Trusted hardware and secure multi-party computation
- User data and model access dictated by smart contract
- Expose highly available micro-services to users and other smart contracts
- Security proof: Universal Composability



Cheng et al., Confidentiality-preserving smart contrat

Democratization of AI: Blockchain of intelligent smart contracts



Key questions to address

Future of ML Systems & Security

How to better understand what security means for AI, learning systems?

How to detect when a learning system has been fooled/compromised?

How to build more resilient learning systems with stronger guarantees?

How to build privacy-preserving learning systems?

How to democratize AI?