# *Formal Analysis in Security*

Jin-Song Dong

Professor, National University of Singapore, dongjs@comp.nus.edu.sg

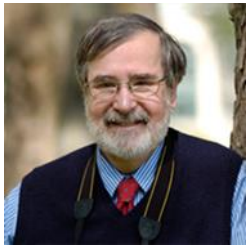
Research Professor, Griffith University, j.dong@griffith.edu.au

1. Process Analysis Toolkit (PAT)
http://pat.comp.nus.edu.sg/

2. Trusted Machine Learning (Silas)
www.depintel.com

# History of Formal Methods

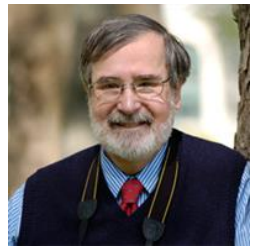
- a 1949 paper “Checking Large Routine” presented by Alan Turing at a conference on High Speed Automatic Calculating Machines at Cambridge University.
- a 1967 paper by R. W. Floyd (1976 Turing Winner) “Assigning meanings to programs” In Proc. Symp. In Applied Mathematics.
- a 1969 paper by C. A. R. Hoare (1980 Turing Winner). “An axiomatic basis for computer programming.” Communications of the ACM. Another great contribution from Hoare is CSP (Communicating Sequential Process)
- Edmund Clarke (CMU) has won 2007 Turing Award and Franklin Institute 2014 Bower Award for his contribution in model checking.



# Our works are based on:

- 1978 C. A. R. Hoare (1980 Turing Winner). Communicating Sequential Processes: event based calculus for modelling concurrency and communication.

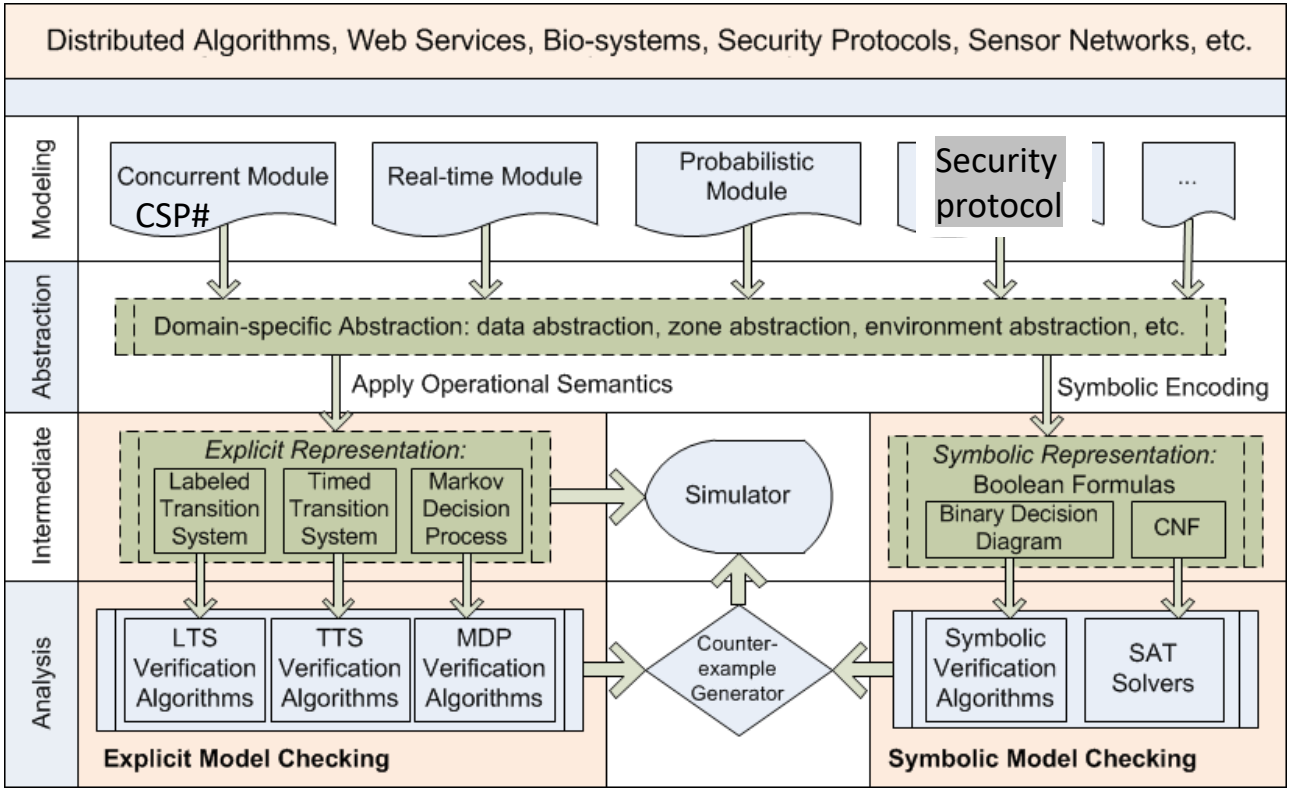
- 2007 three researchers won Turing Award for inventing model checking, one of them E. Clarke (CMU) has also won Franklin Institute 2014 Bower Award.
  - successfully applied in industries, e.g.,




# Past 11 years work on Process Analysis Toolkit (PAT)

## (CAV '09'11'13'16, ICSE'08'13'15'16, FM'09'11'12'16, TSE'08'13'15'17'18)

http://pat.comp.nus.edu.sg/




Dong, NUS

Sun, SUTD

Liu, NTU

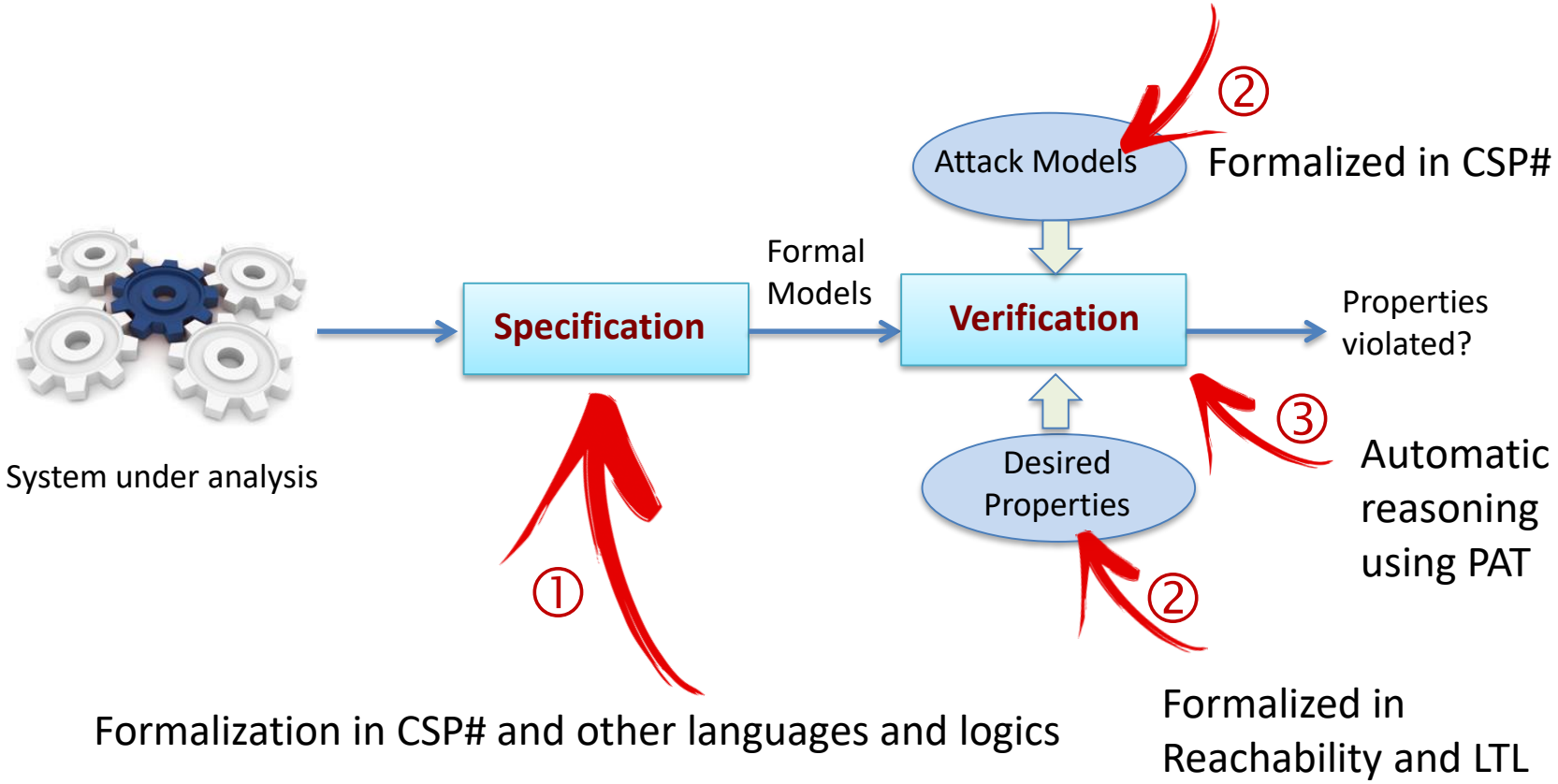
- 1Million lines of C# code, 15 verification systems, 200+ build in examples, 12 PhDs, 100+ publications
- 4000+ registered users from 900+ organizations in 150 countries, e.g. Huawei, HP, Sony, Mitsubishi, NTT, Toyota …
- 20 Year ICFEM Most Influential System Award (2018)

20 Year ICFEM Most Influential System Award
PRESENTED TO
Jin Song Dong, Yang Liu, Jun Sun
"For developing the Process Analysis Toolkit (PAT) [ICFEM'08-'17] used by 4000+ registered users from 900+ industry and organisations"
14 November 2018
Date
Founder and 20 Year ICFEM Award Chair

# Current/new projects with PAT (2017 – 2022)

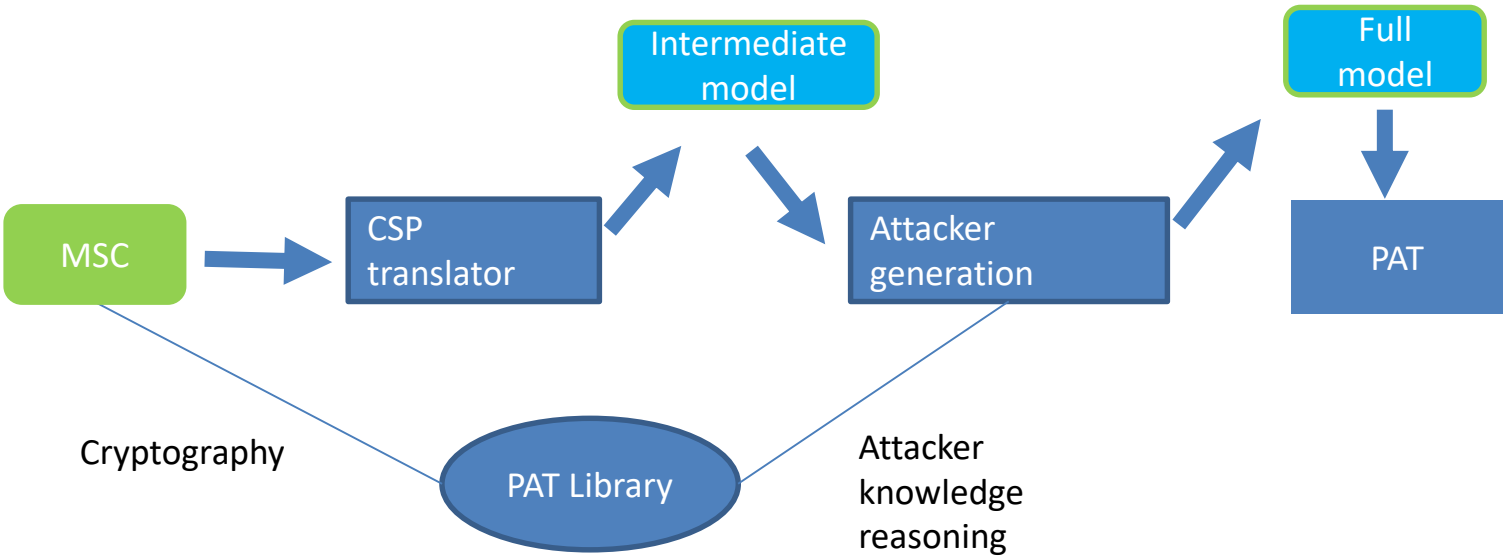
- Singapore-UK joint cyber security project on smart grid security and privacy
  - With Andrew Martin (Oxford) and Bai Guangdong (Griffith)
- Securify: A Compositional Approach of Building Security Verified System ($7M)
  - With Liu Yang (NTU), Sun Jun (SUTD), Chin Weingan (NUS) etc
- Trustworthy systems from untrusted Components ($7M)
  - With Abhik, Prateek (NUS) etc
- Singtel-NUS Cyber Security joint lab ($43M).
  - With David Rosenblum, Liang Zhenkai (NUS), etc
- Blockchain verification using timed and game-theory based reasoning
  - With Muthu (Griffith), Jun, Yang, Guangdong etc
- Trusted Machine Learning (Verified and Explainable AI)
  - With Zhe, Hadrien (Griffith) etc
- Trusted Autonomous Systems (reasoning with unknown/limited sensing data)
  - With Zhe, Hadrien (Griffith) and Mahony and Oxenham (Aus Defense) etc
- Sports Analytics (strategy analysis & match fixing investigation with sports data)
  - With Zhe, Hadrien and Jie (Griffith) etc

# A General Approach of Formal Analysis Using Formal Methods



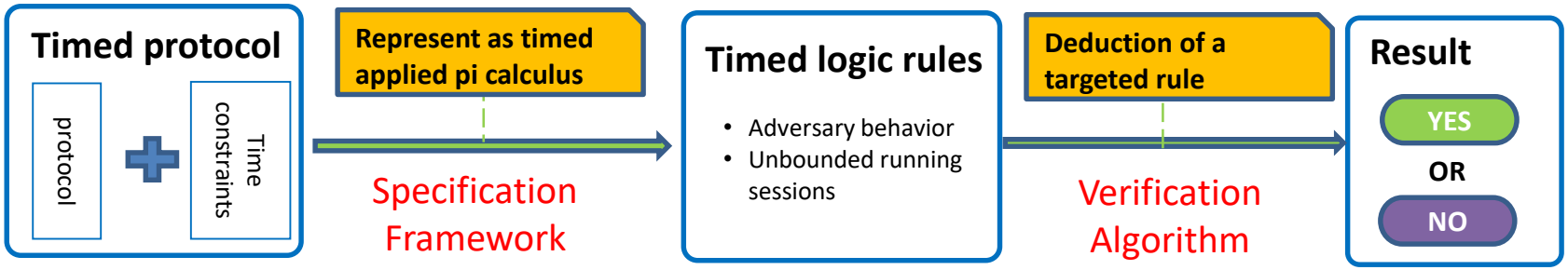
# Security Protocol Module

- PAT library for cryptographic primitives and attack reasoning
- Automatic attack generator that adds attack process



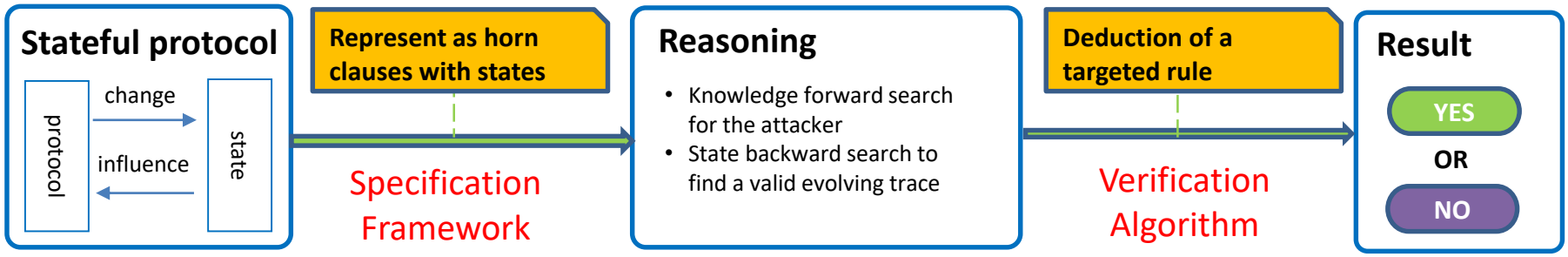
# Timed Security Protocol

- Propose timed calculus to specify timed security protocols
- Develop verification algorithms to verify timed security protocols
- Found a time related attack in the Kerberos V protocol

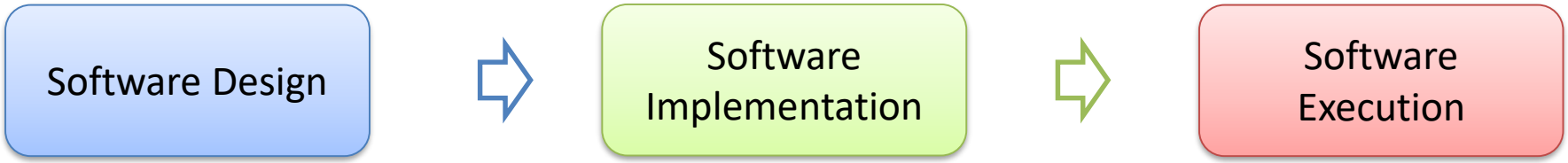


# Stateful Security Protocol

- Stateful security protocols - global states which influence the protocol behavior and may unboundedly evolving
- Developed a specification framework based on horn clauses
- Developed a verification algorithm for verifying stateful security protocols with unbounded evolving of global states



# Using PAT for Secure Software Analytics




- Trusted Computing
- Smart Grid
- Authentication
- Blockchain

- Payment
- Web Authentication
- IoT (Smart Home)
- Android

- Android
- Mobile Authentication
- Smart Grid

# Using PAT to Analyse Trusted Computing Systems

- Propose a formal foundation and framework (TrustFound) for model checking trusted computing systems
- Including a spectrum of threat models
- Found 6 implicit assumptions and 2 severe logic flaws in systems using TPM (Trusted Platform Module)
- Supported by joint research grant from Singapore and UK
  - Collaborators: Prof. Andrew Martin (Oxford) and Dr. Guangdong Bai (Griffith University)

# Using PAT to Analyse Blockchain Consensus Algorithms

- Build model of Blockchain using CSP# with 10 models to simulate several attacks
  - Attackers who overwrite blocks (minority, half and majority)
  - Attackers who submit invalid blocks (minority, half and majority)
  - Censorship attackers (minority, half and majority)

- Verify the Tendermint consensus algorithm using PAT
  - Using probabilistic model checking to analyse the fraction of malicious nodes

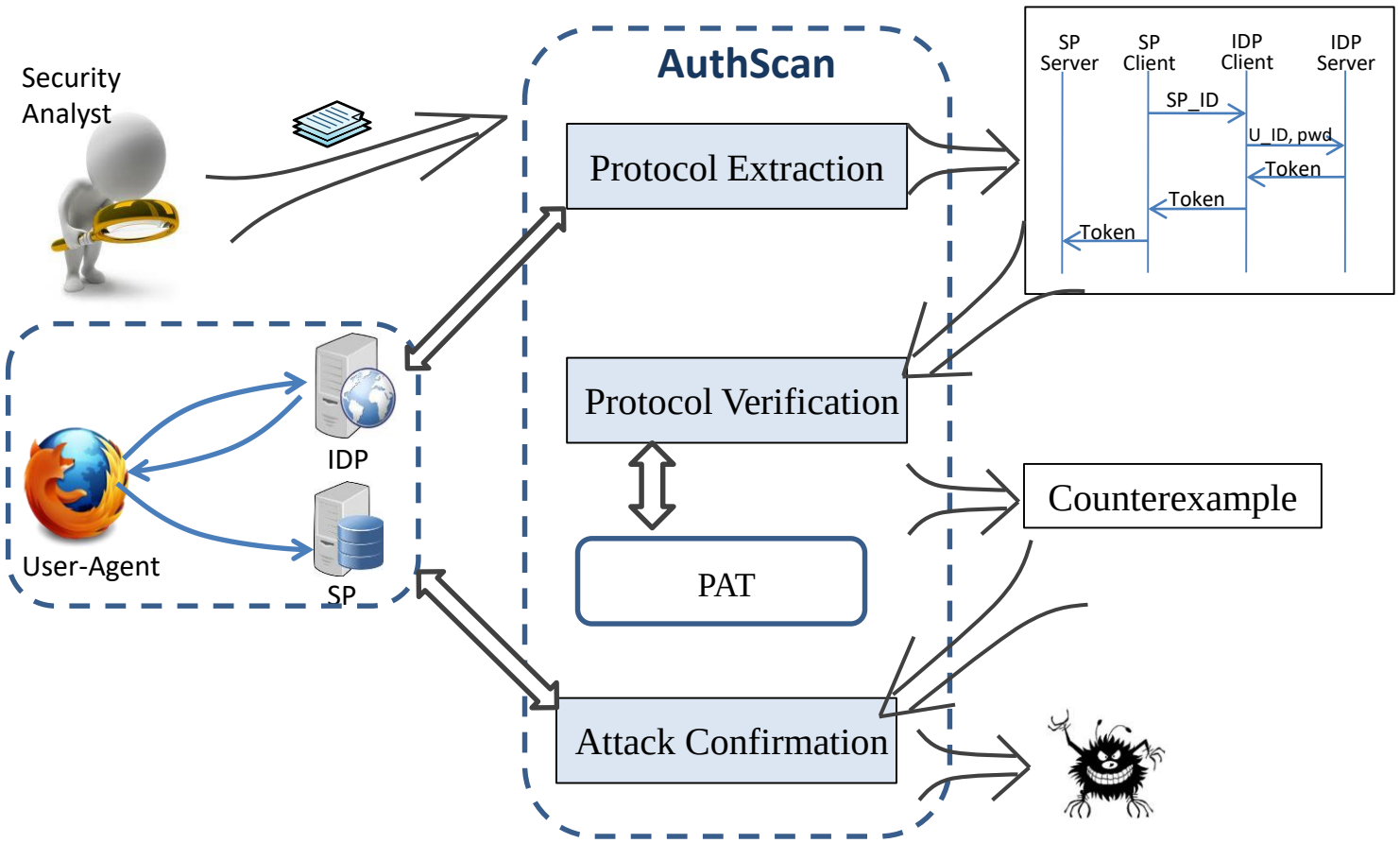
- Analyse five properties
  - Deadlockfree-ness
  - Ability to reach consensus
  - Immunity against block overwrites
  - Immunity against Invalid blocks
  - Immunity against Censorship attacks

# Using PAT to Analyse Web Authentication Systems

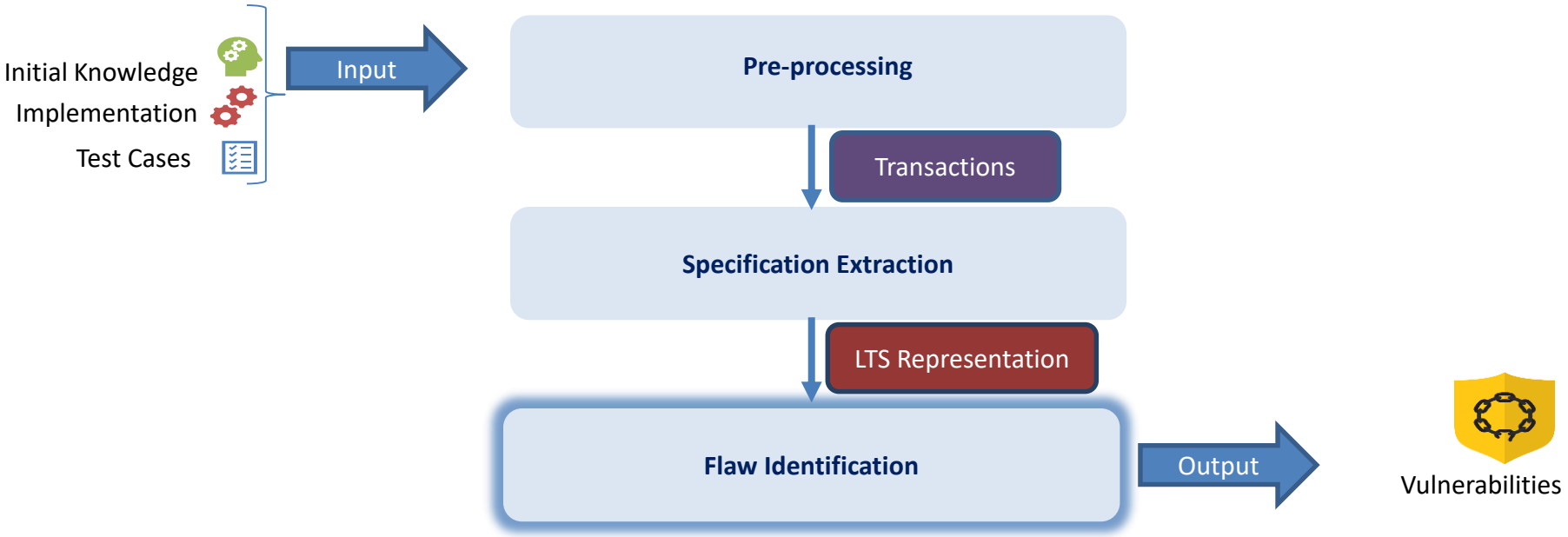
- Propose an end-to-end framework (AuthScan)
  - extracts protocols from implementations
  - uses PAT to check extracted protocols for vulnerabilities

- Finds 7 real-world security vulnerabilities in Web Authentication


# Overview of AuthScan



# Analyse Internet of Things



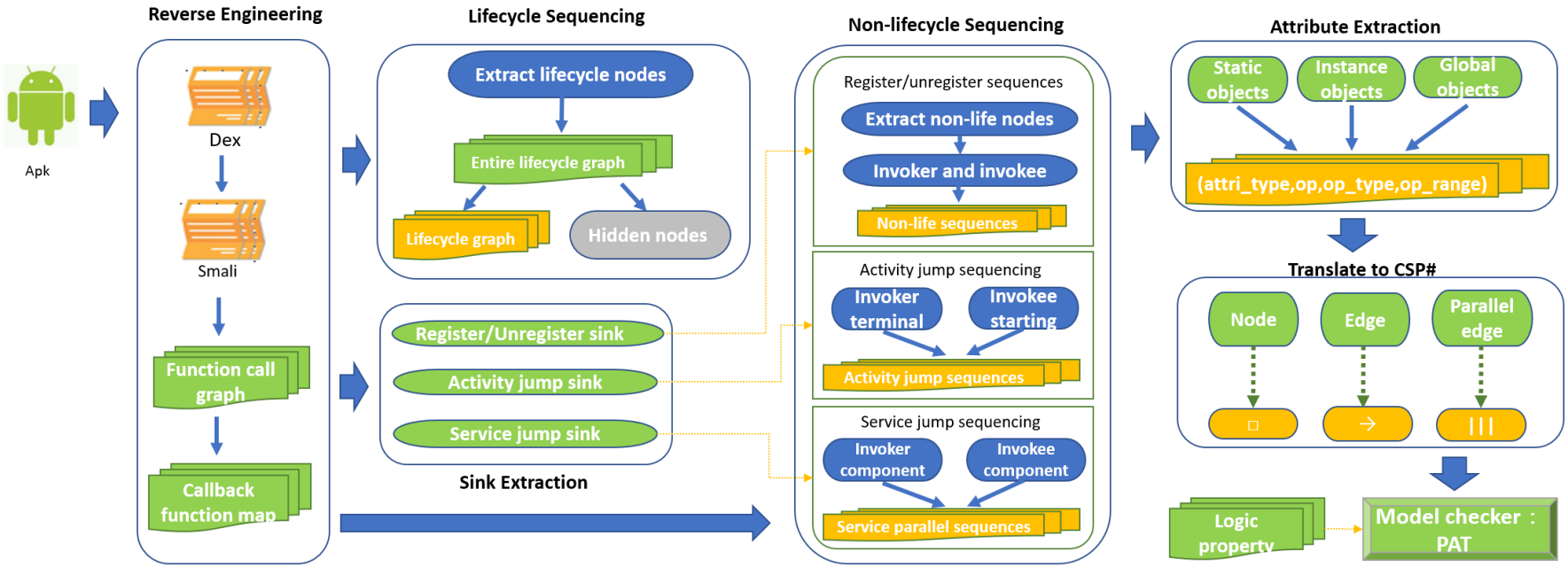

- 15 real-world security vulnerabilities in smart home systems

PHILIPS hue personal wireless lighting

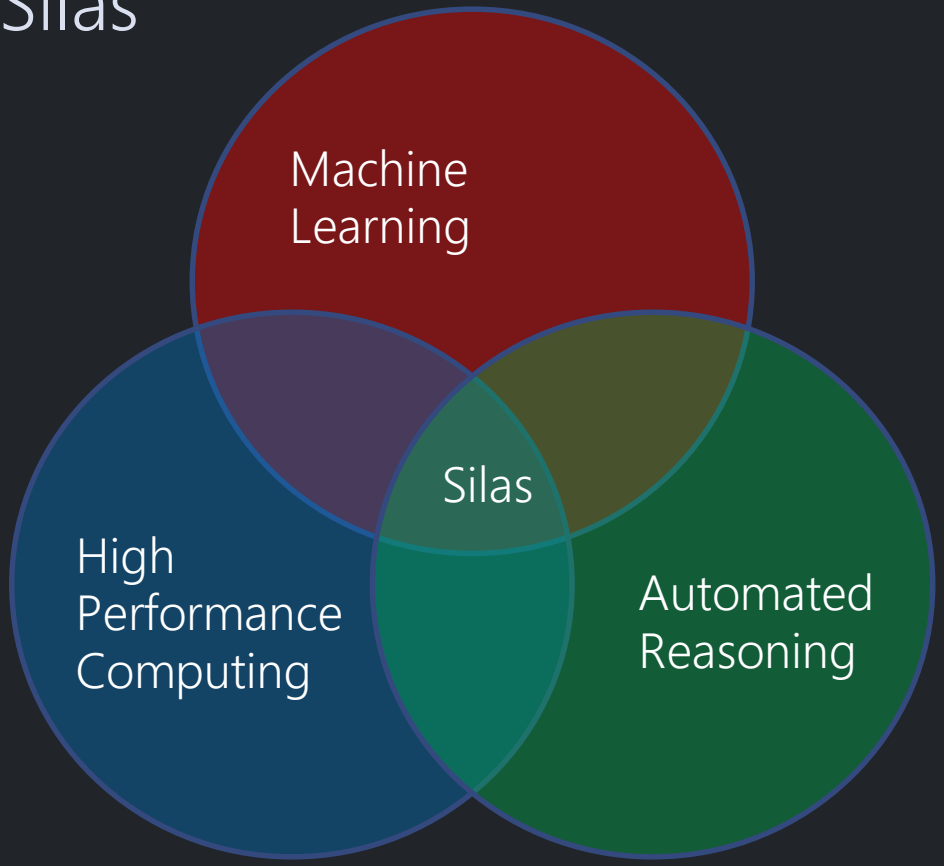
LIFX

# Callback graph verification

- Automatically extract triggering relations between components
- Automatically modelling and verifying the relations using PAT



# Introducing Silas




https://www.depintel.com/

# Silas: Trusted Machine Learning Tool

https://www.depintel.com/

- **Model Insight** adopts automated reasoning technique to provide insights on model prediction.
- **Model Audit** uses formal verification techniques to mathematically prove that the prediction model conforms with the user's specifications.

## Models that make sense

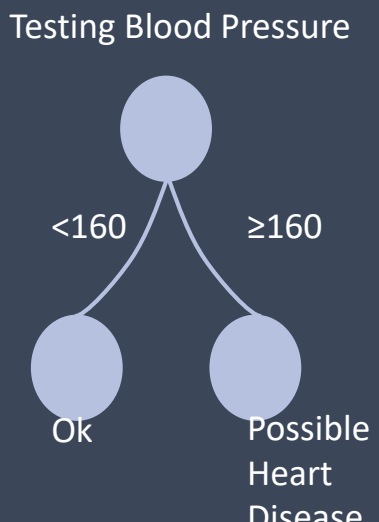
Unlike many neutral network and deep learning techniques which generate models that are hard to interpret, Silas produces models based on decision-trees, which are easy to be understood.

## Memory Friendly

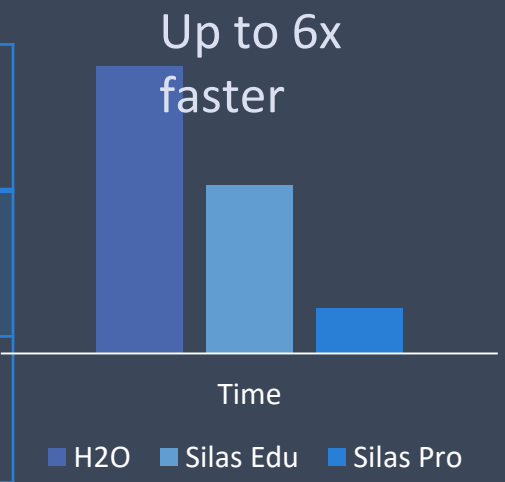
Silas is built to be memory efficient. In extreme cases, Silas will even swap data from memory to hard drive so that machines with moderate memory can still perform computation.

## Speed of The Art

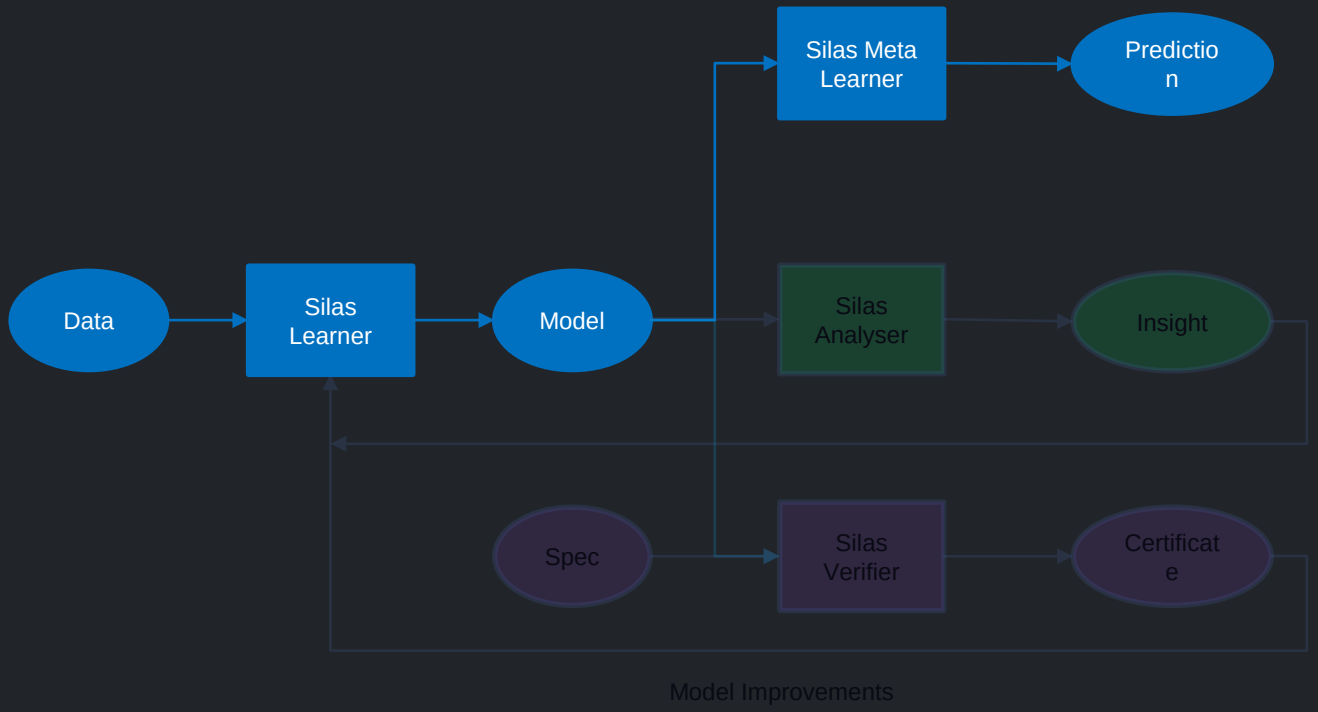
Silas uses Intel's Thread Building Blocks technology and very low level code optimisation to maximise the speed of concurrent computation.



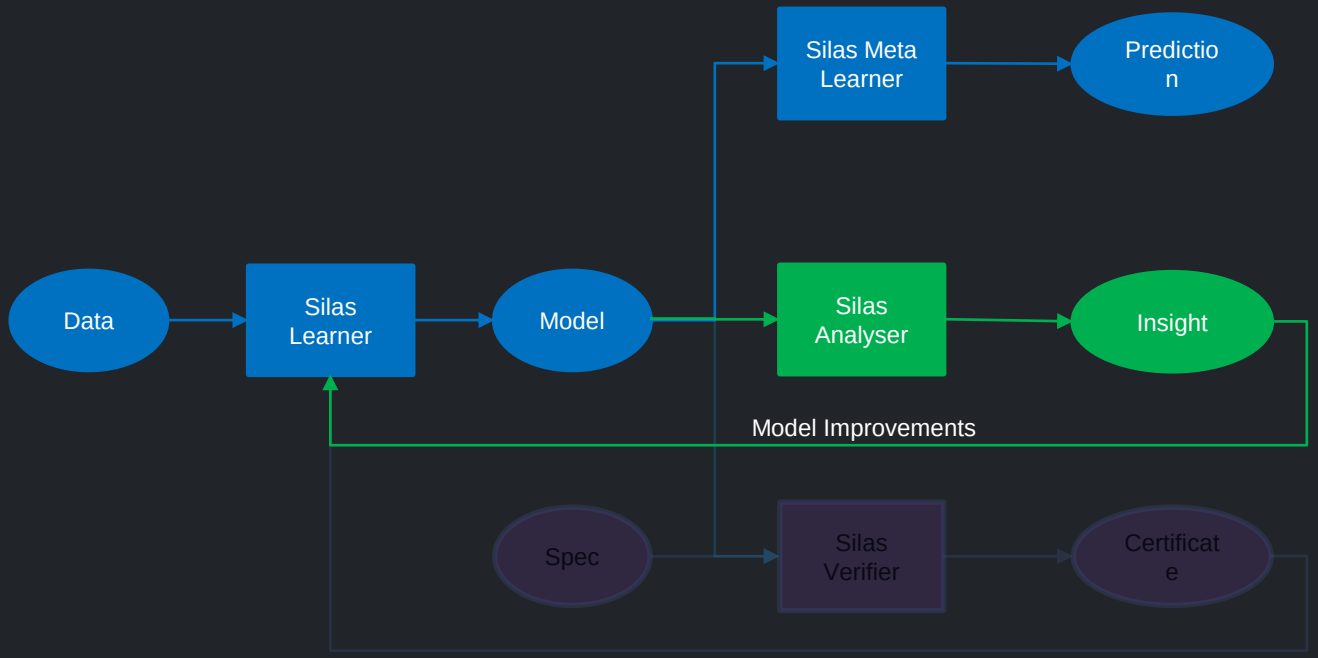

| Data set size | R | Python | H2O | Spark | Silas Edu | Silas Pro |
|---|---|---|---|---|---|---|
| 1M instances | Crash | 20GB | 5GB | Crash | **4GB** | **1.4GB** |
| 10M instances | Crash | Crash | 25GB | Crash | **20GB** | **9.8GB** |



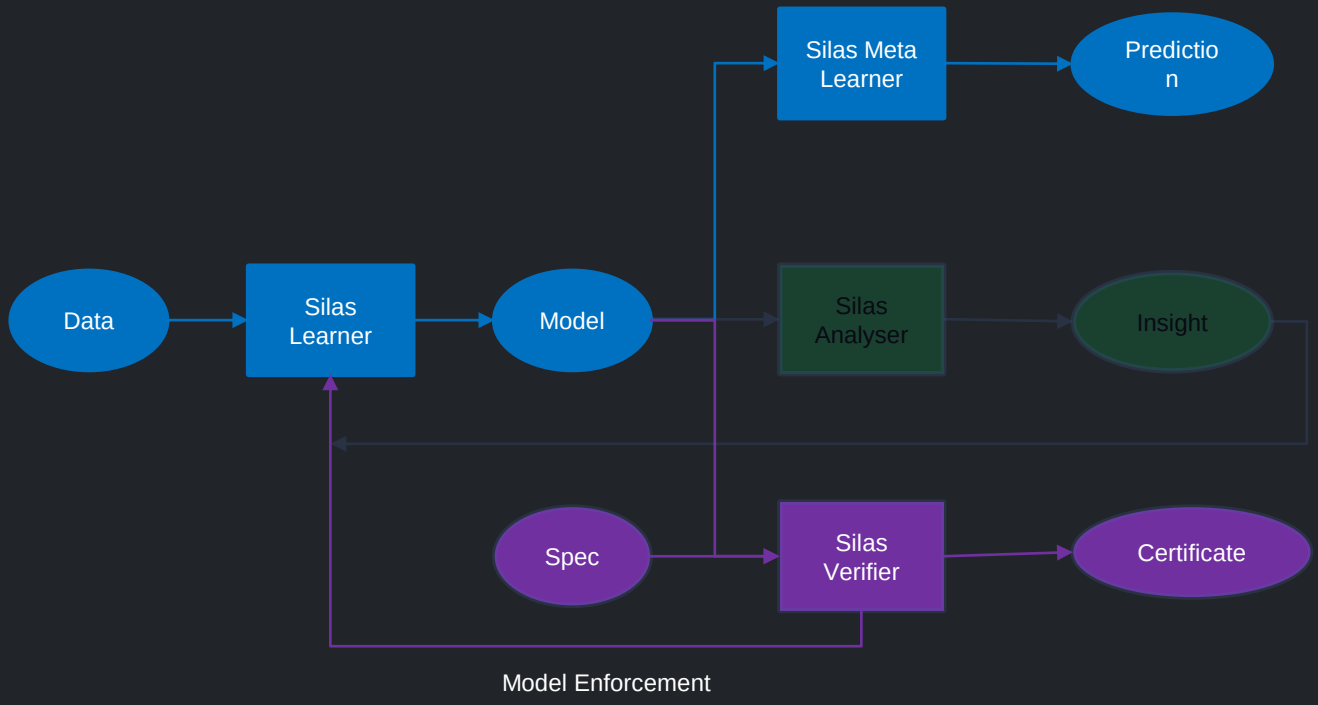
# Silas Machine Learning



# Silas Model Insight



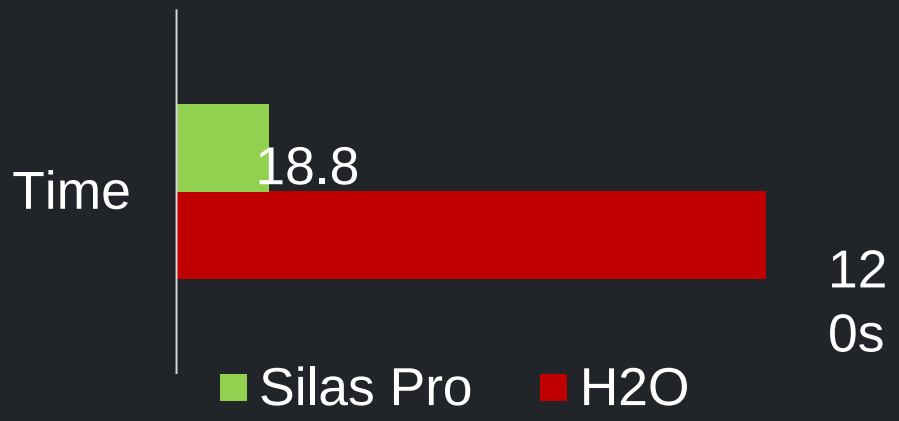
# Silas Model Audit



# High Performance Computing – Speed

To train a model of similar size that yields similar predictive performance,

Silas Pro is **6.3x faster** than the best tools.



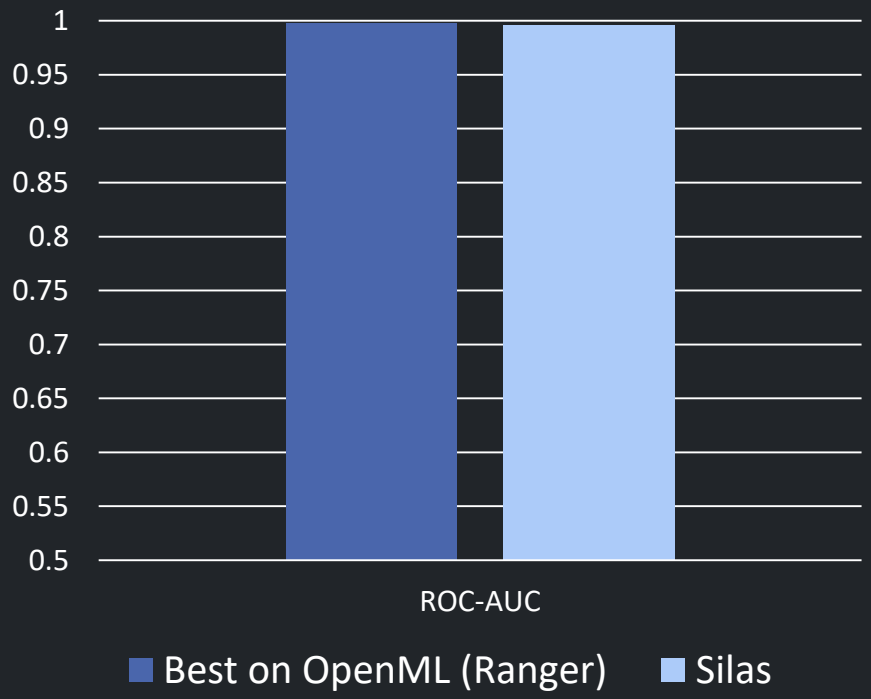
# High Performance Computing – Memory Usage

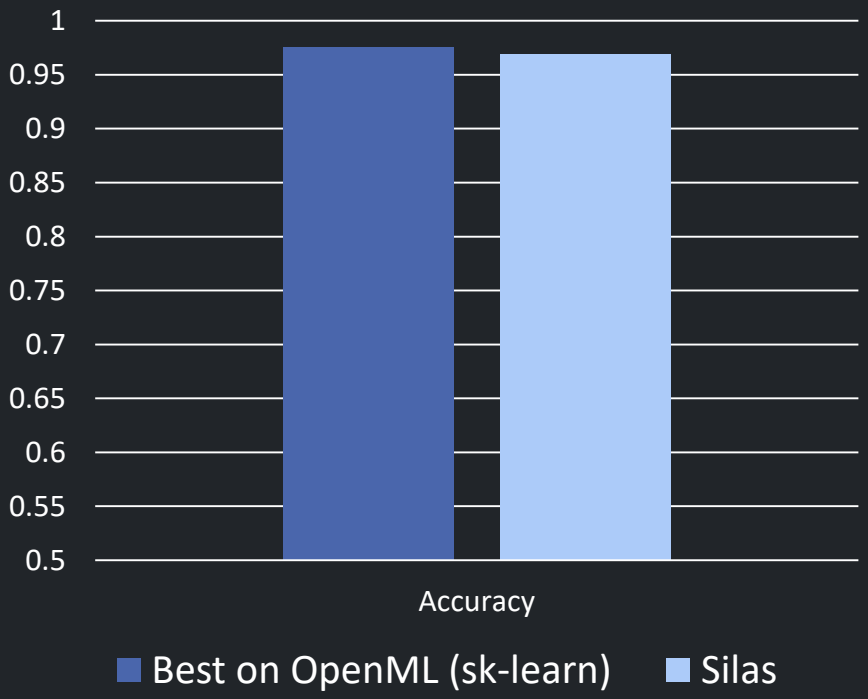
Silas Pro uses at least **60% less resources** than competitors.

| Data set size/RAM usage | R | Python | H2O | Spark | Silas Pro |
|---|---|---|---|---|---|
| 1M instances | Crash | 20GB | 5GB | Crash | **1.4GB** |
| 10M instances | Crash | Crash | 25GB | Crash | **9.8GB** |

# Predicting Phishing Websites with Silas

https://www.openml.org/d/4534 Size: 11K data points. Classification with 10-fold cross validation.



We are looking for collaborators

Thank you!

# Extra slides:

# Formula Extraction

- Node formula: Logical formulae with arithmetic, comparison, membership etc.
- Branch Formula: Conjunction of every node on a branch implies a class.
- Tree Formula: Disjunction of the branch formulae.
- Each formula has a weight.
- May need sampling for analysis:
  - Sample a subset of branches in a tree.
  - Sample a subset of trees in a forest.

# AI Failures

- Complex AI stock trading software caused a trillion dollar flash crash.
- Microsoft chatbot Tay became racist and bigoted.
- Amazon’s Alexa offered porn to child.
- Self-driving car had a deadly accident.
- AI designed to predict crime acted racist.
- …

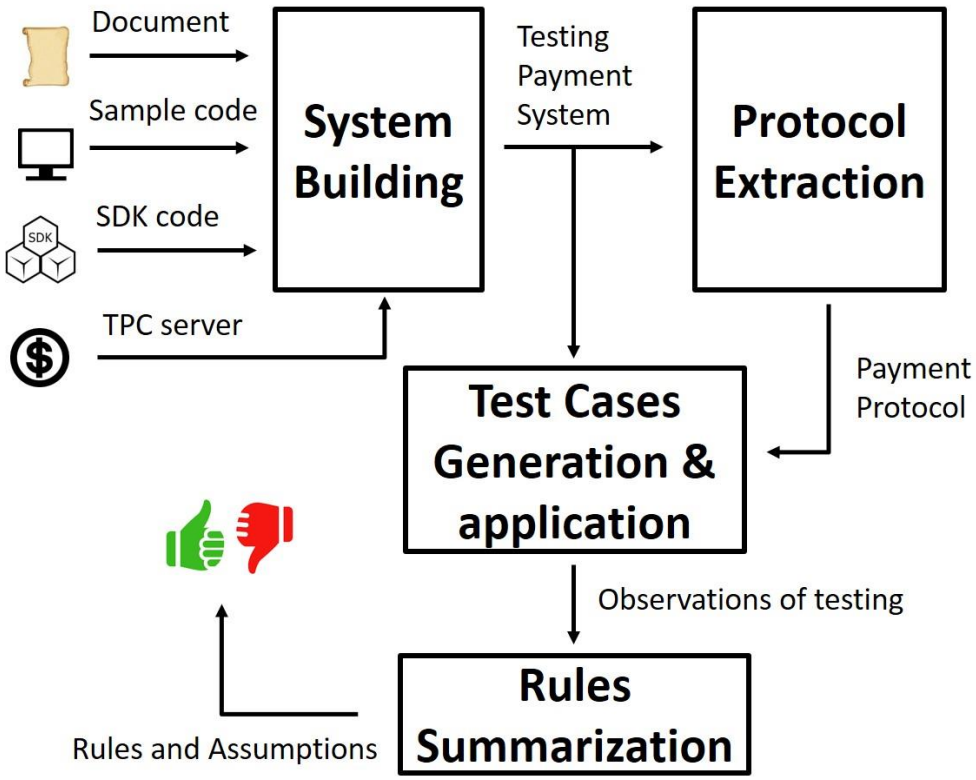
# Model Analysis

- Minimal Unsatisfiable Core (MUC)
  - Inconsistencies among the decision trees.
  - Use the MUC to improve the trees.
  - Use the MUC as weak classifier in boosting.
- Maximum Satisfiable Subset (MSS)
  - The common ground of the decision trees.
  - Use the MSS to find key features.
  - Use the MSS to explain the rationale of the decision making.

# Model Verification

- User-defined constraints
  - E.g., popUpWidnow = 1 implies positive phishing website.
- Use SMT solving to verify the model against constraints.
  - F is a tree formula.
  - C is a user-defined constraint.
  - Check satisfiability of ~(F -> C). If unsat, then C is valid in F.
- Model Enforcement.
  - Build decision trees that are guaranteed to satisfy user-defined constraints.
  - And have good predictive performance.

# Analyse Web Payment Protocols




Find a security issue in PayPal SDK

# Outcomes of Analysing Payment protocols and IoT systems

- 1 real-world security issue in payment protocols, 3 bugs from apps using PayPal


- 15 real-world security vulnerabilities in smart home systems



# Overview of TrustFound

- TrustFound: a framework for model checking trusted platforms