SMPC for Decentralized Distributed Systems

Frederic Klein – Final Talk

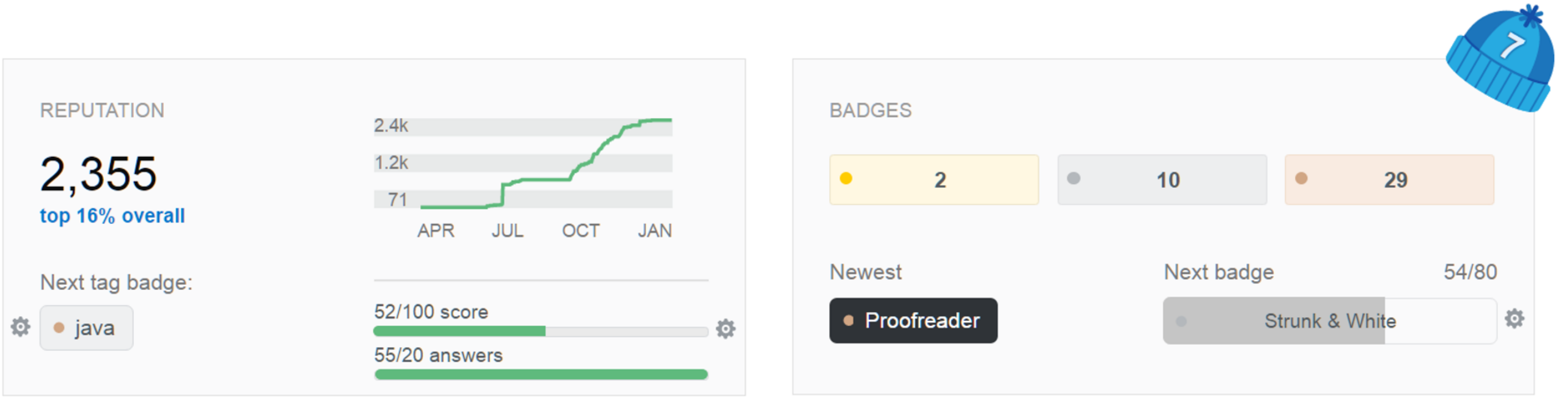
Institute of Medical Informatics
Uniklinik RWTH Aachen

Overview

- Motivation
  - Gamification
  - Secure Multi-Party Computation
  - Hygiene Games
- Design
  - SPAN on Android
  - Distributed system: coordinator election, clock synchronization, database consistency
  - Decentralized system: decentralized SMPC, distributed database
- Evaluation
  - Simulation
  - Android Integration (demonstration)
- Discussion
- Outlook

Gamification

- Generate intrinsic motivation
  - Among other motivators: competition and social comparison
- Examples
  - Stackoverflow, Amazon, runtastic, etc.



REPUTATION

2,355

top 16% overall

Next tag badge:

● java

52/100 score

55/20 answers

BADGES

● 2 ● 10 ● 29

Newest

● Proofreader

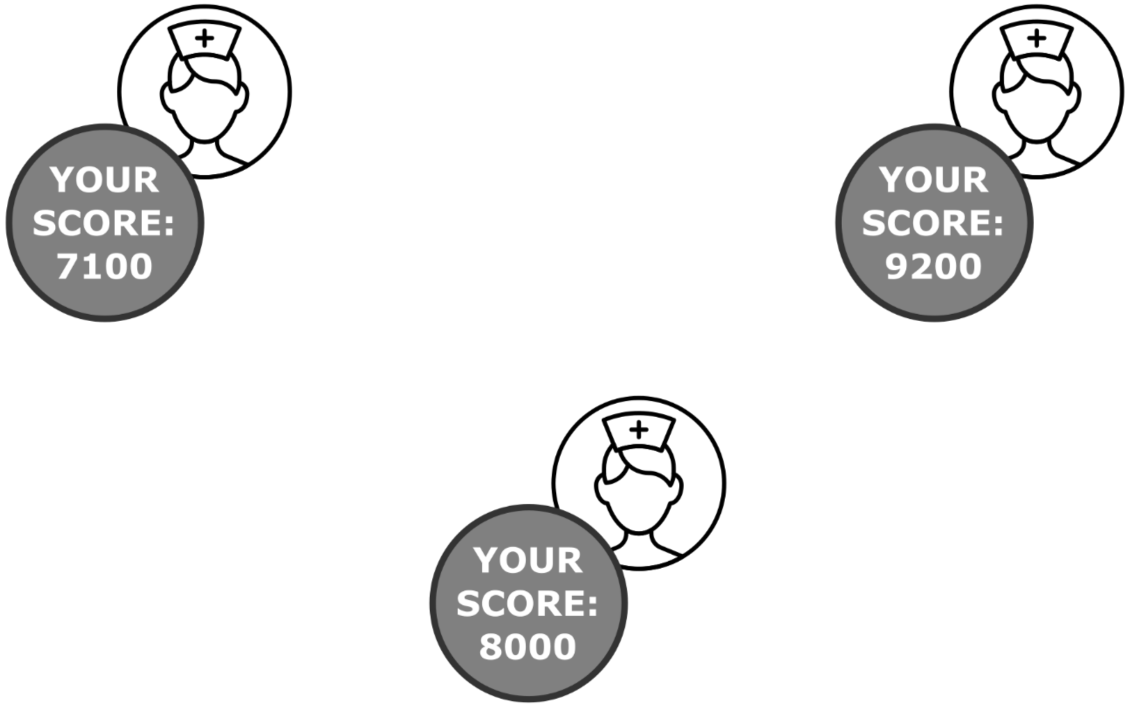
Next badge 54/80

● Strunk & White

Privacy Concerns

- Sensitive data
  - Sharing might result in disadvantage
- Example: Hygiene Games
  - Gamification approach for hand-hygiene compliance
  - Targeting health-care professionals
  - High privacy demands
  - Independent from Internet access

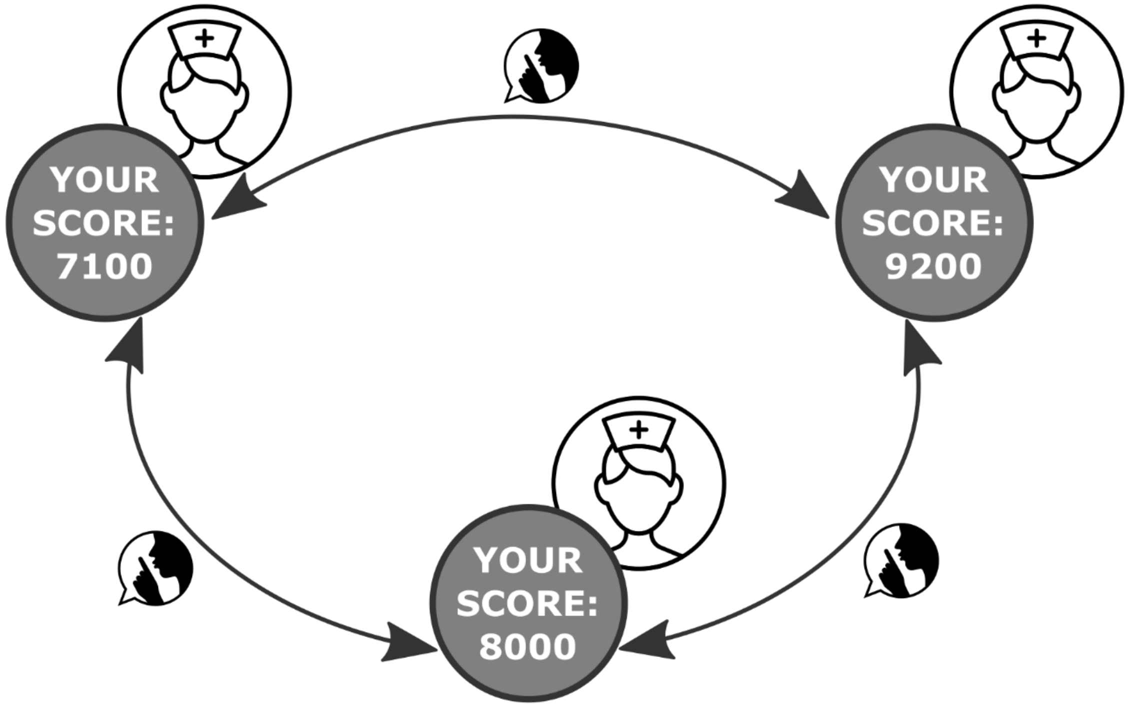
Privacy Protection



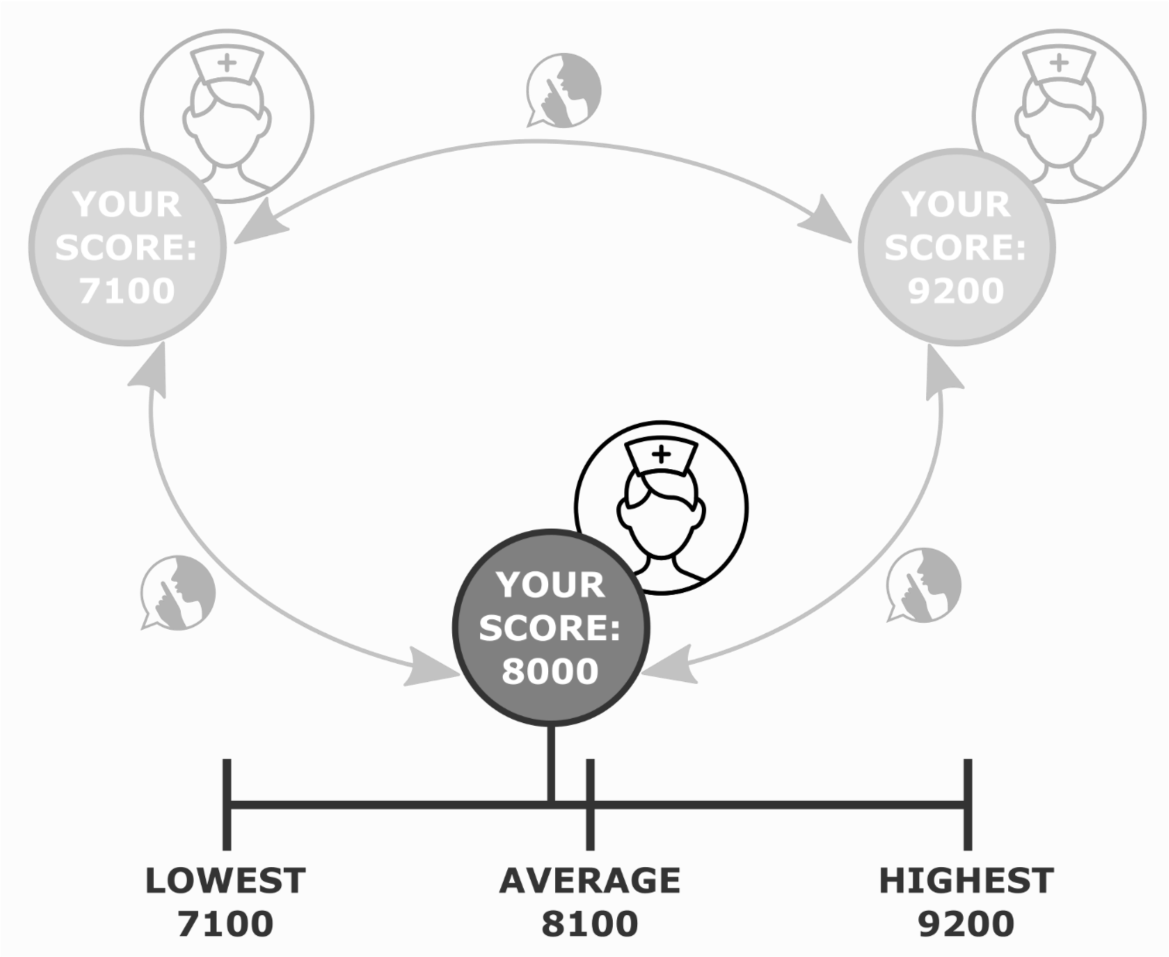
- Personal data on own device
- Modest value without comparison

Privacy Protection

- Exchange data for comparison



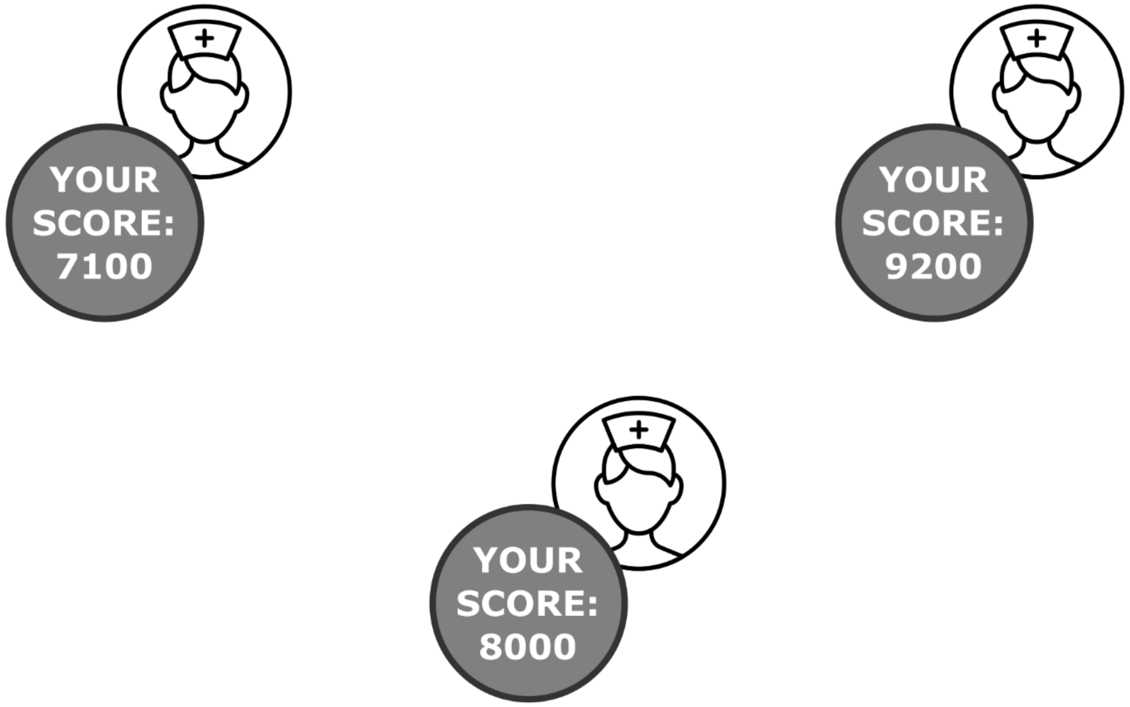
Privacy Protection



SMPC

- Subfield of cryptography:
  - compute function over inputs of multiple parties
  - keep the inputs private

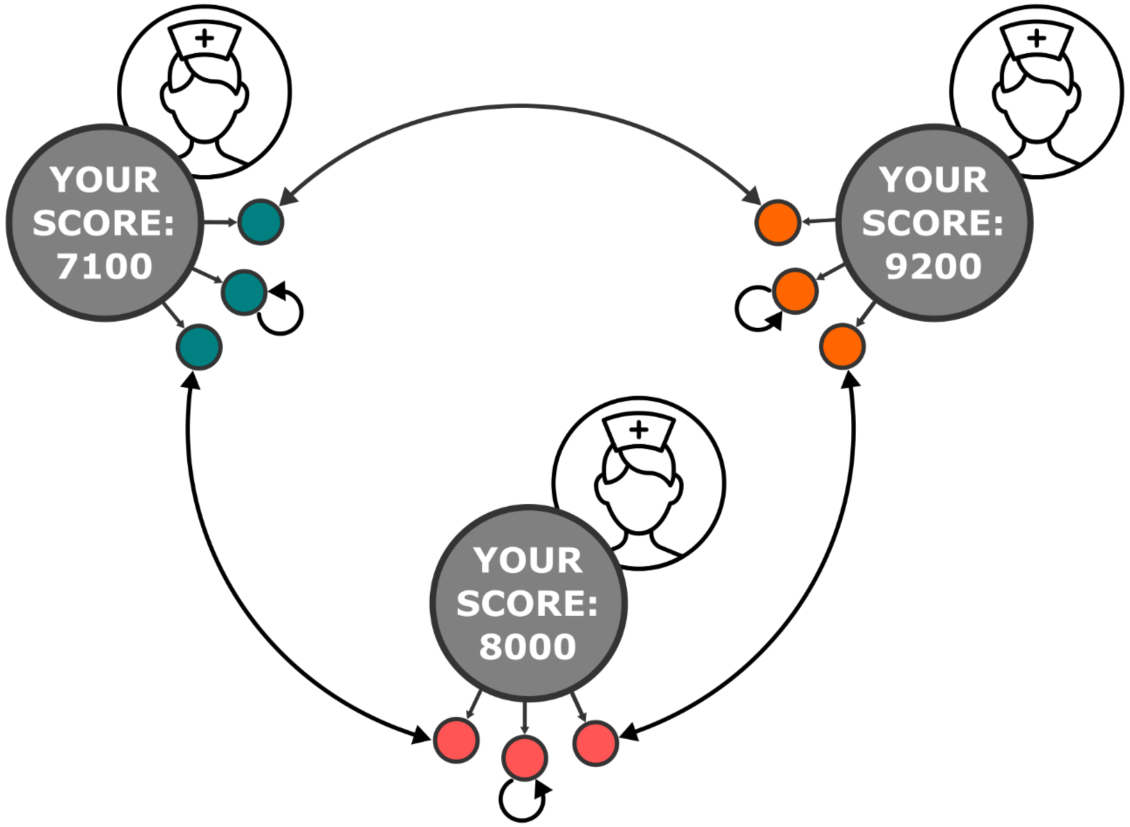
SMPC



- Three parties
- Score as input

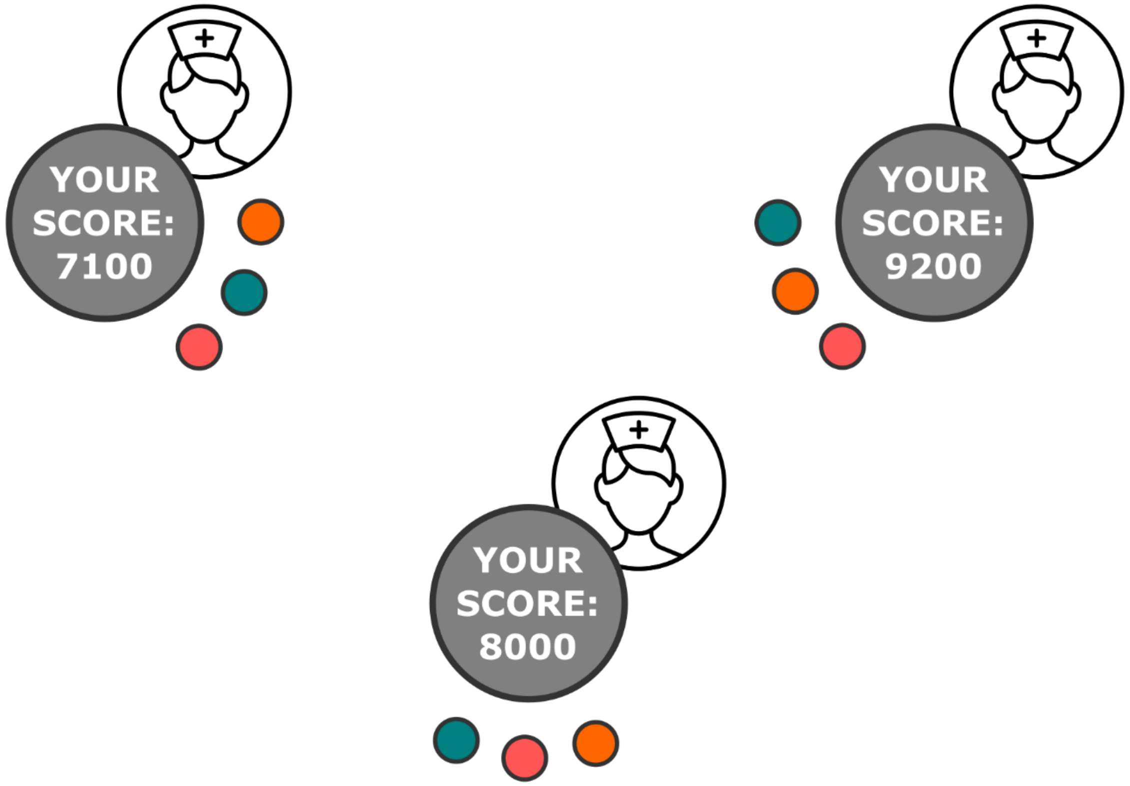
SMPC

- Secret sharing: n shares for n parties



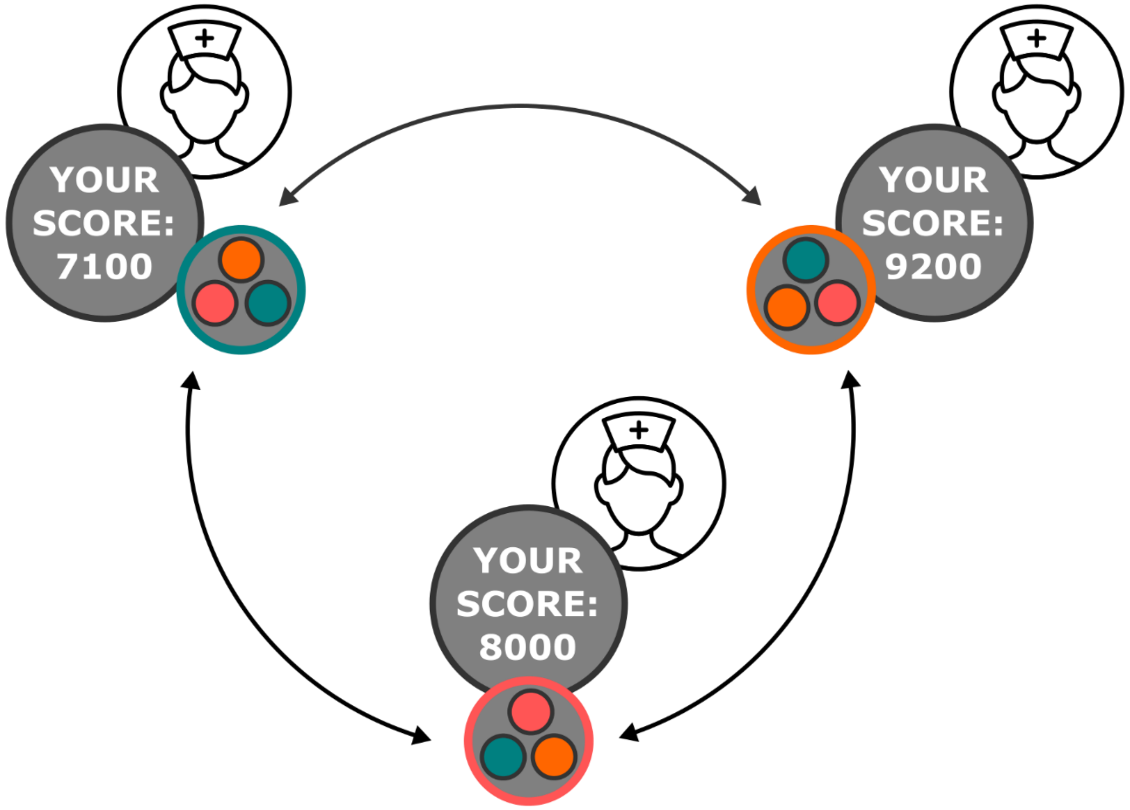
SMPC

- Each player: set of n shares



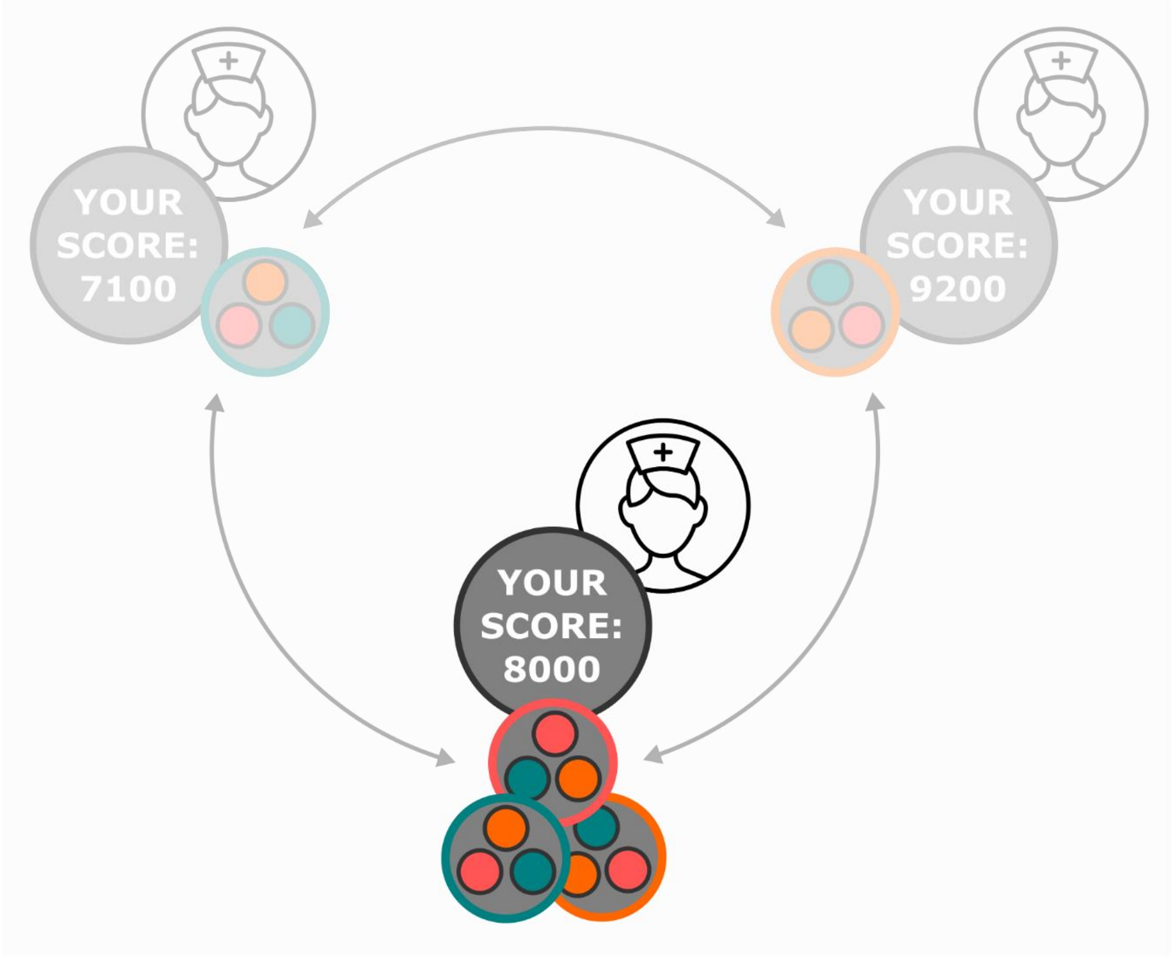
SMPC

- Computation on shares
- Broadcasting of result



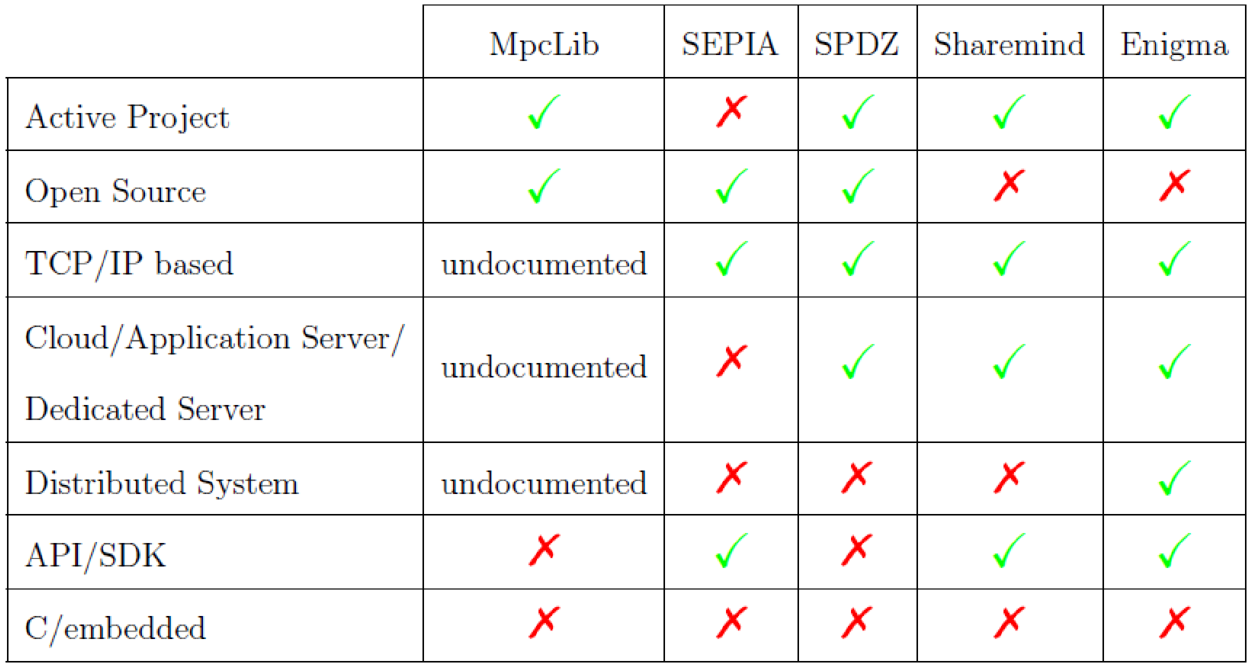
SMPC

- Each party:
  - Complete information for computation
  - Other inputs remain secret



Existing Frameworks

- Existing frameworks
  - Rely on the Internet
  - Clinical environment: EMI
  - Powerful but complex



	MpcLib	SEPIA	SPDZ	Sharemind	Enigma
Active Project	✓	✗	✓	✓	✓
Open Source	✓	✓	✓	✗	✗
TCP/IP based	undocumented	✓	✓	✓	✓
Cloud/Application Server/ Dedicated Server	undocumented	✗	✓	✓	✓
Distributed System	undocumented	✗	✗	✗	✓
API/SDK	✗	✓	✗	✓	✓
C/embedded	✗	✗	✗	✗	✗

Requirements

- Framework
  - System-wide statistics using SMPC
  - Infrastructure-less, self-forming Mobile ad-hoc network
  - Feasible algorithms for acceptance

Design: MANETs

- MANET/SPAN
  - Mesh networks with moving nodes
  - Infrastructure-less, self-forming, self-healing

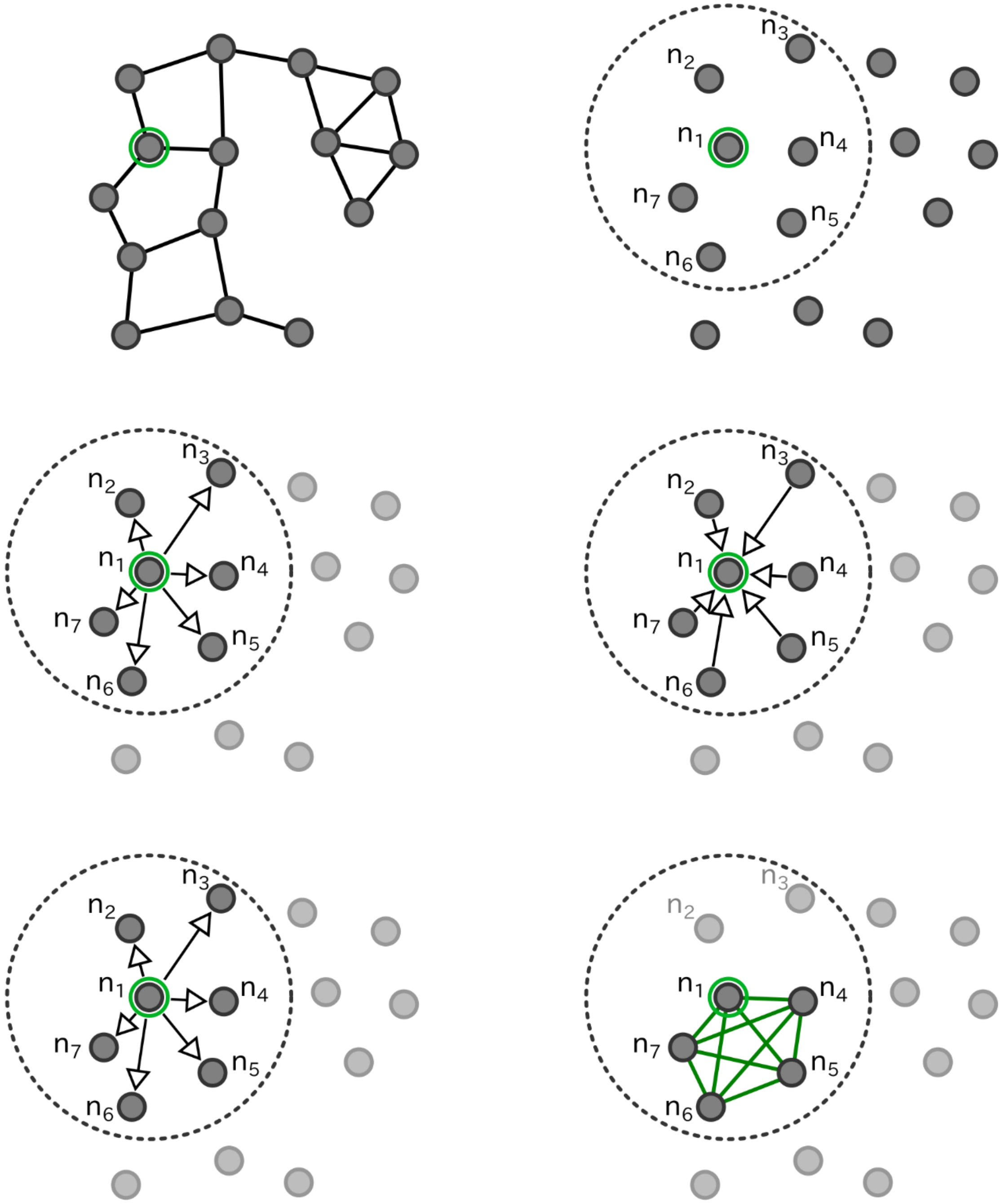
TODO: FIGURE

Design: MANETs

- MANET/SPAN on Android
  - Not included in API
  - Abilities of Bluetooth modules vary
    - Sequential communication
  - Pairing-less connection
    - Insecure RFCOMM
    - Encryption layer

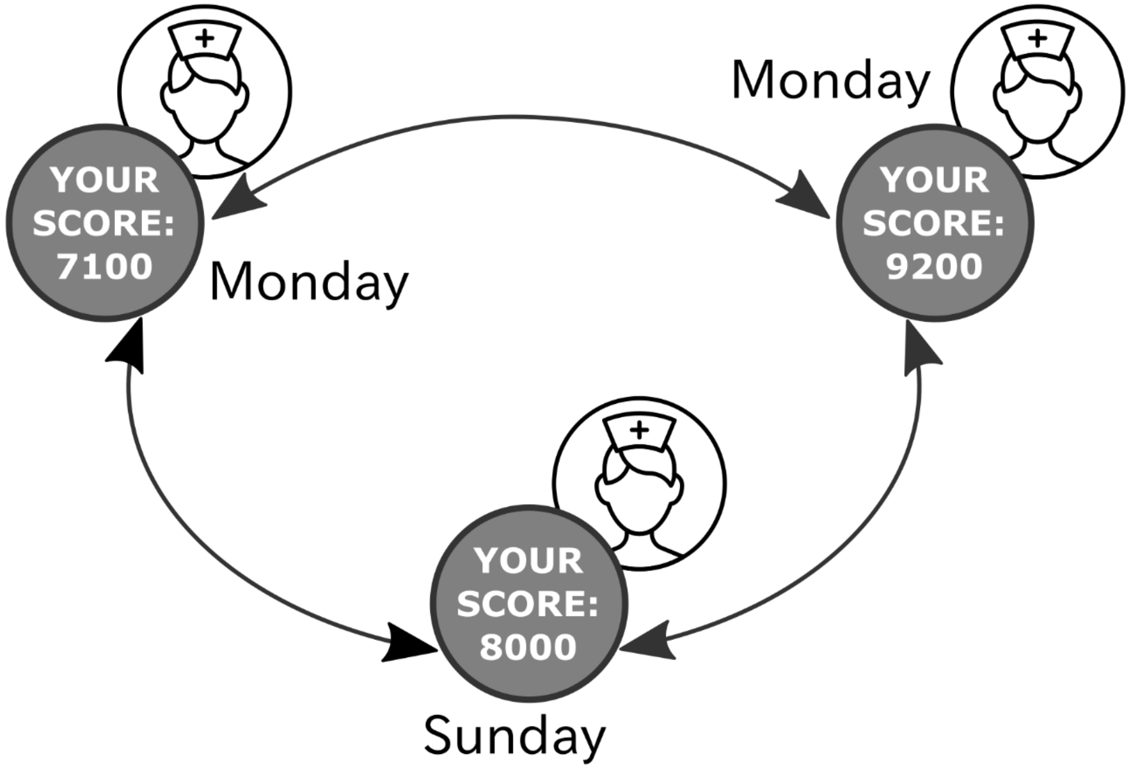
Design: Coordinator Election

- Coordinator election
  - Event driven
  - Timer based



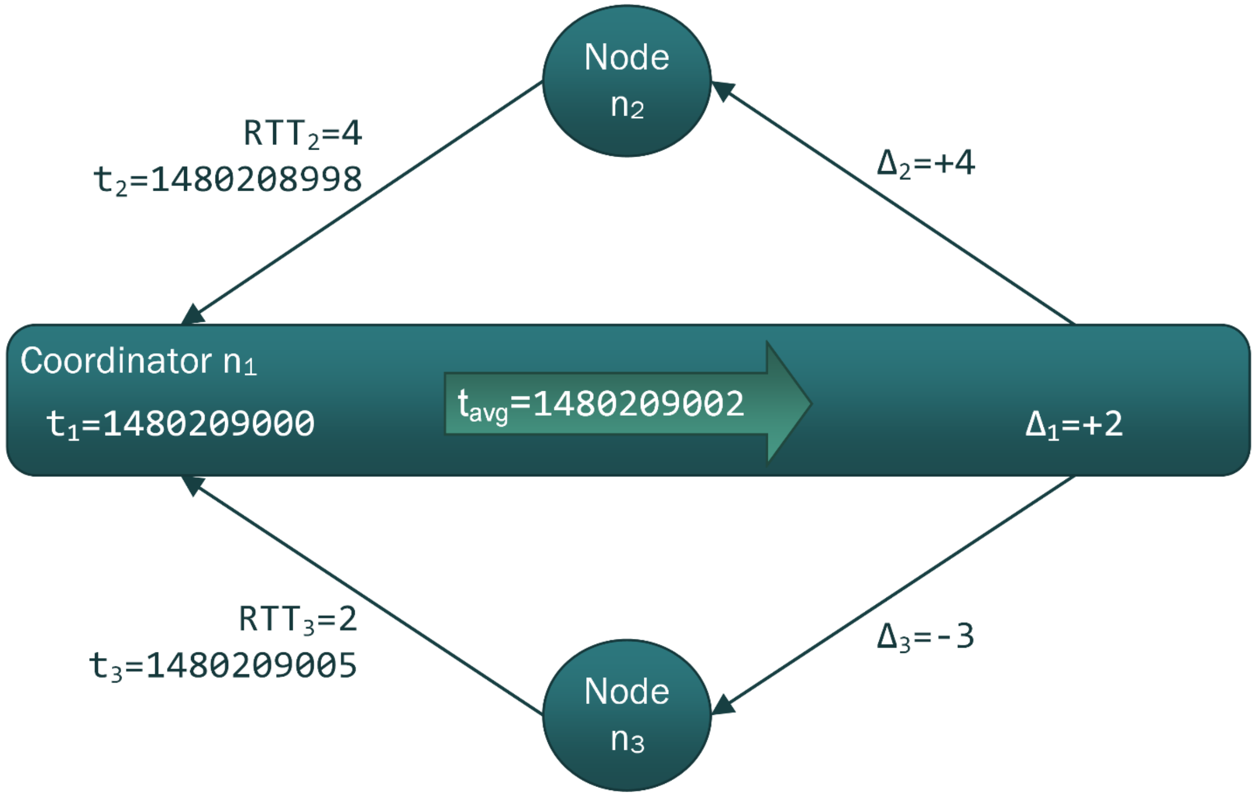
Design: Clock Synchronization

- Internal synchronization of clocks
  - Berkeley Algorithm



Design: Clock Synchronization

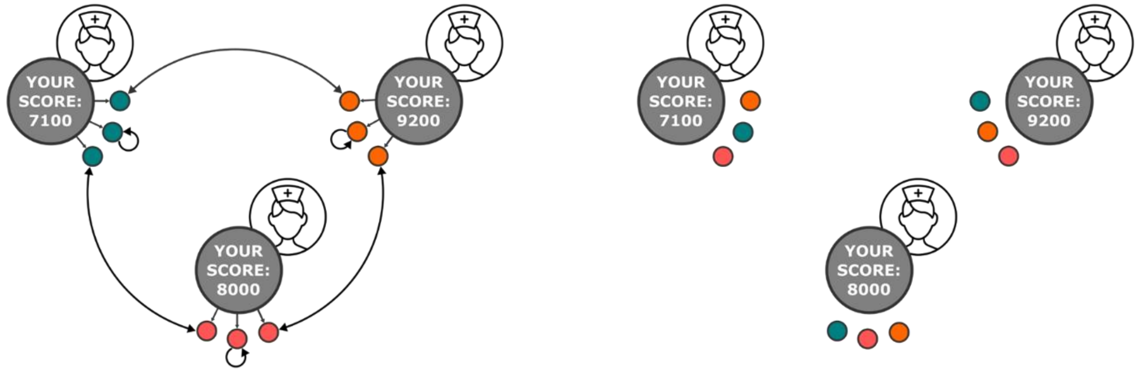
- Internal synchronization of clocks
  - Berkeley Algorithm



Design: Distributed Database

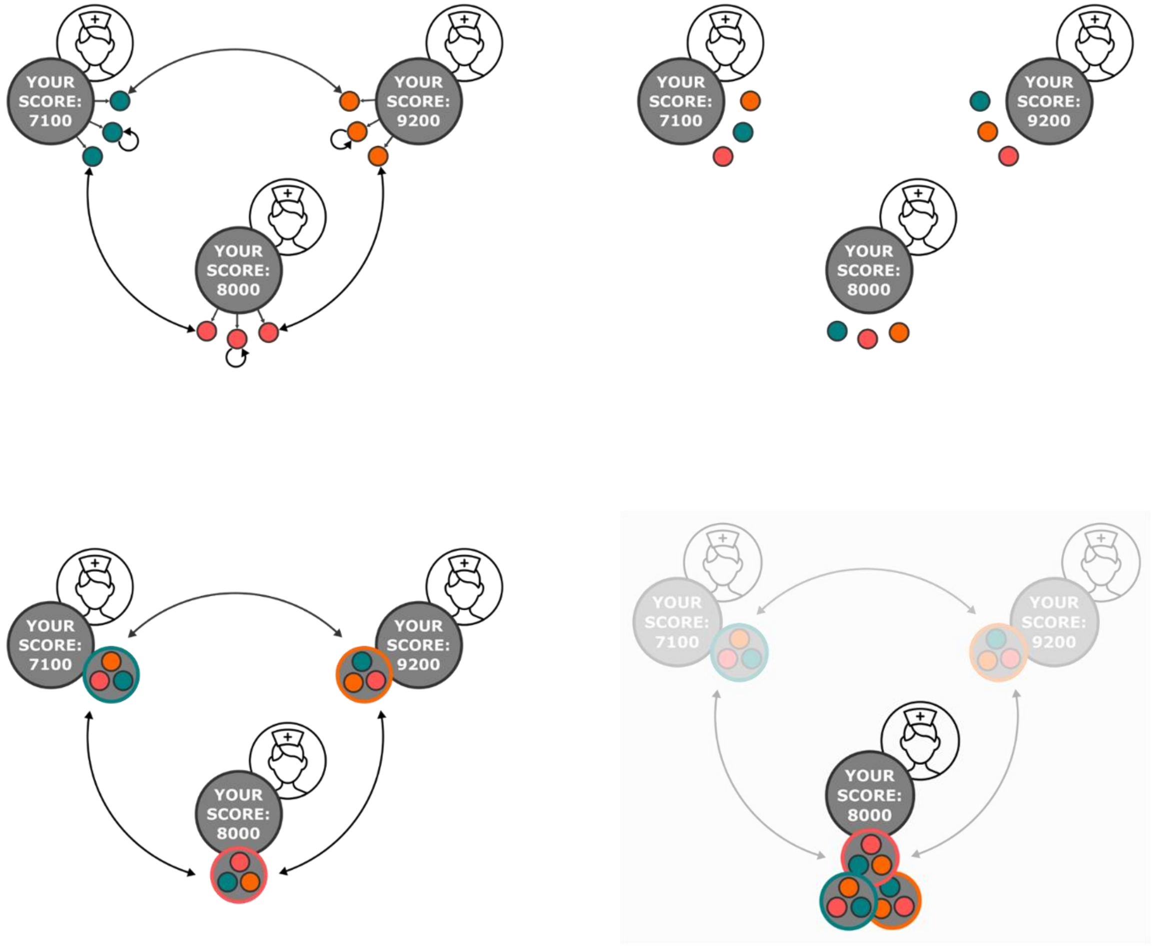
- Node maintains copy of database
  - Callback for framework to query database
  - Database comparison with neighboring nodes
  - Hash-tree for finding differences

Design: SMPC algorithms



- Based on Shamir's secret sharing
  - Defines function for shares
  - Lagrange interpolation for recombination

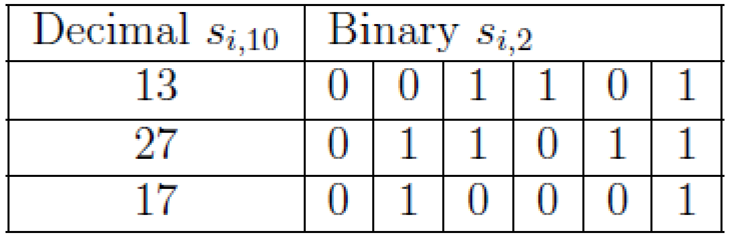
Design: SMPC algorithms



- Secure Sum
    - Sum over secrets equals Lagrange interpolation over sum of shares

Design: SMPC algorithms

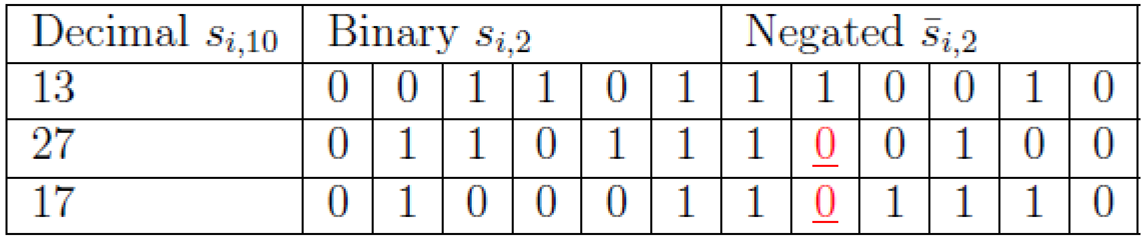
- Secure Maximum
  - Bit decomposition
  - Secure sum for each bit position from MSB
  - Self disqualification



Decimal $s_{i,10}$	Binary $s_{i,2}$					
13	0	0	1	1	0	1
27	0	1	1	0	1	1
17	0	1	0	0	0	1

Design: SMPC algorithms

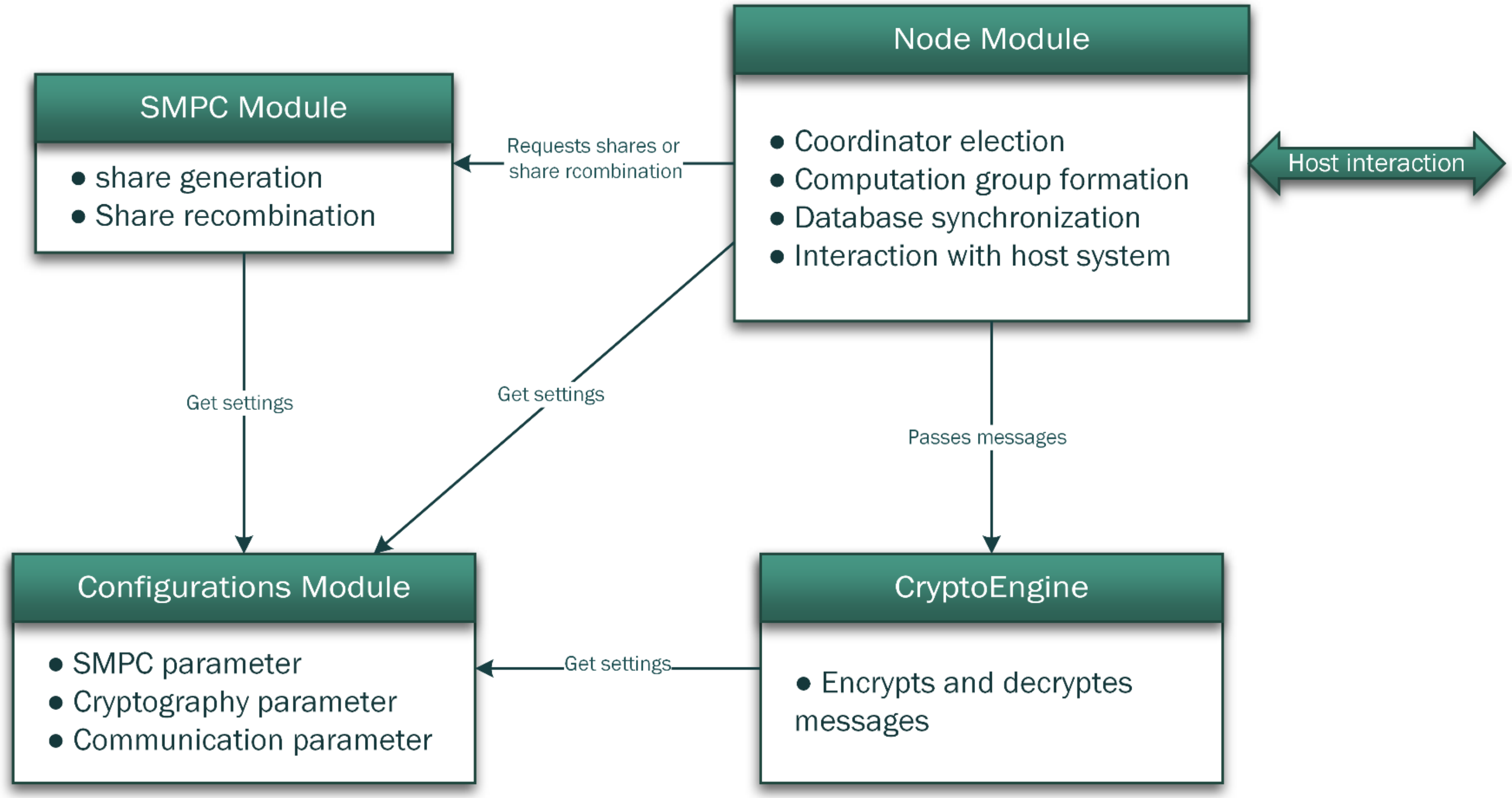
- Secure Minimum
  - Bit decomposition
  - Inverting
  - Secure sum for each bit position from MSB
  - Self disqualification
  - Inverting result



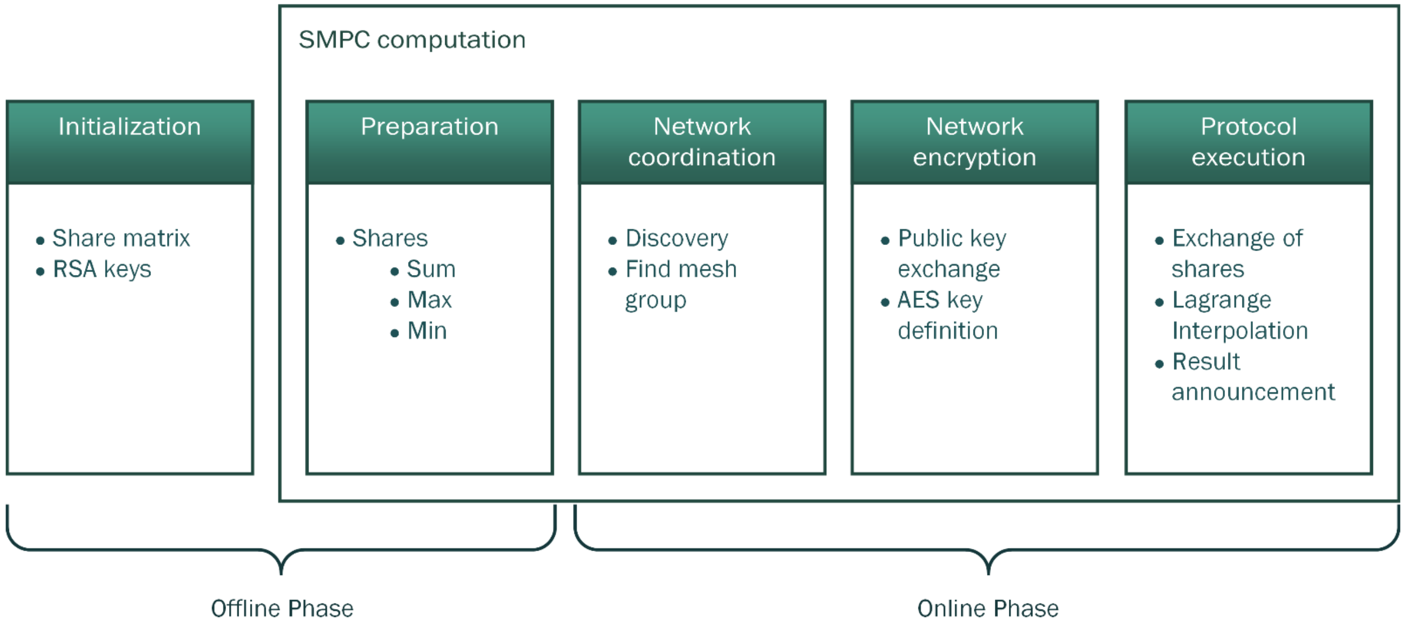
Decimal $s_{i,10}$	Binary $s_{i,2}$						Negated $\bar{s}_{i,2}$					
13	0	0	1	1	0	1	1	1	0	0	1	0
27	0	1	1	0	1	1	1	0	0	1	0	0
17	0	1	0	0	0	1	1	0	1	1	1	0

Design: framework

- C library for compatibility
- All parameters in Configuration
- Interfacing through Node Module



Evaluation



- Performance of online phase
- Android integration
- Implementation of core system
  - Node module, SMPC module, integration of crypto library WolfCrypt

Evaluation: Simulation

- Linux script creating nodes
- Adjustable number of nodes
- Delayed named pipes to simulate wireless connection
- Restriction of CPU through power management

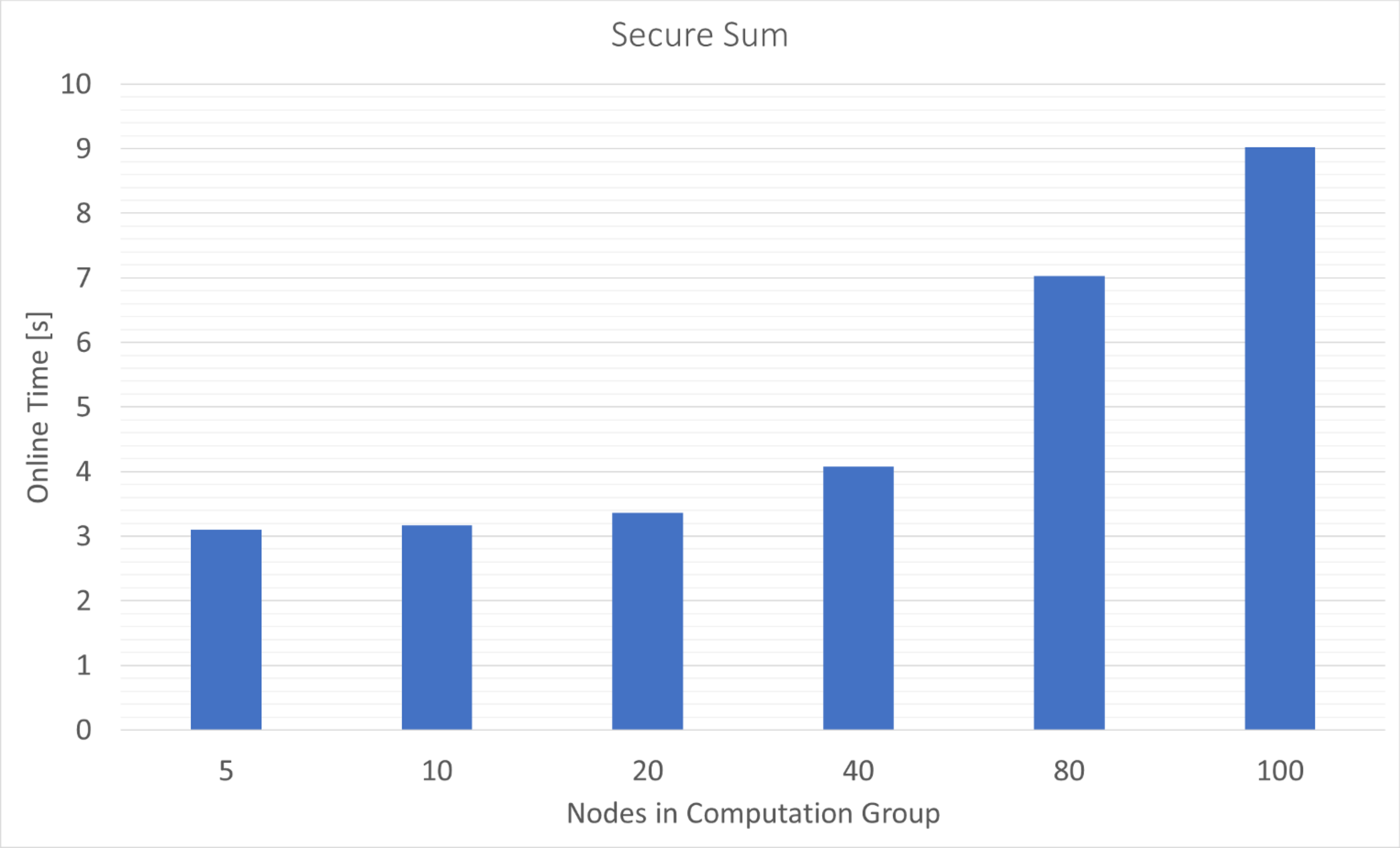
Evaluation: Android Integration

- NDK/JNI wrapper
- Demonstration

TODO:
Smartphone Demo

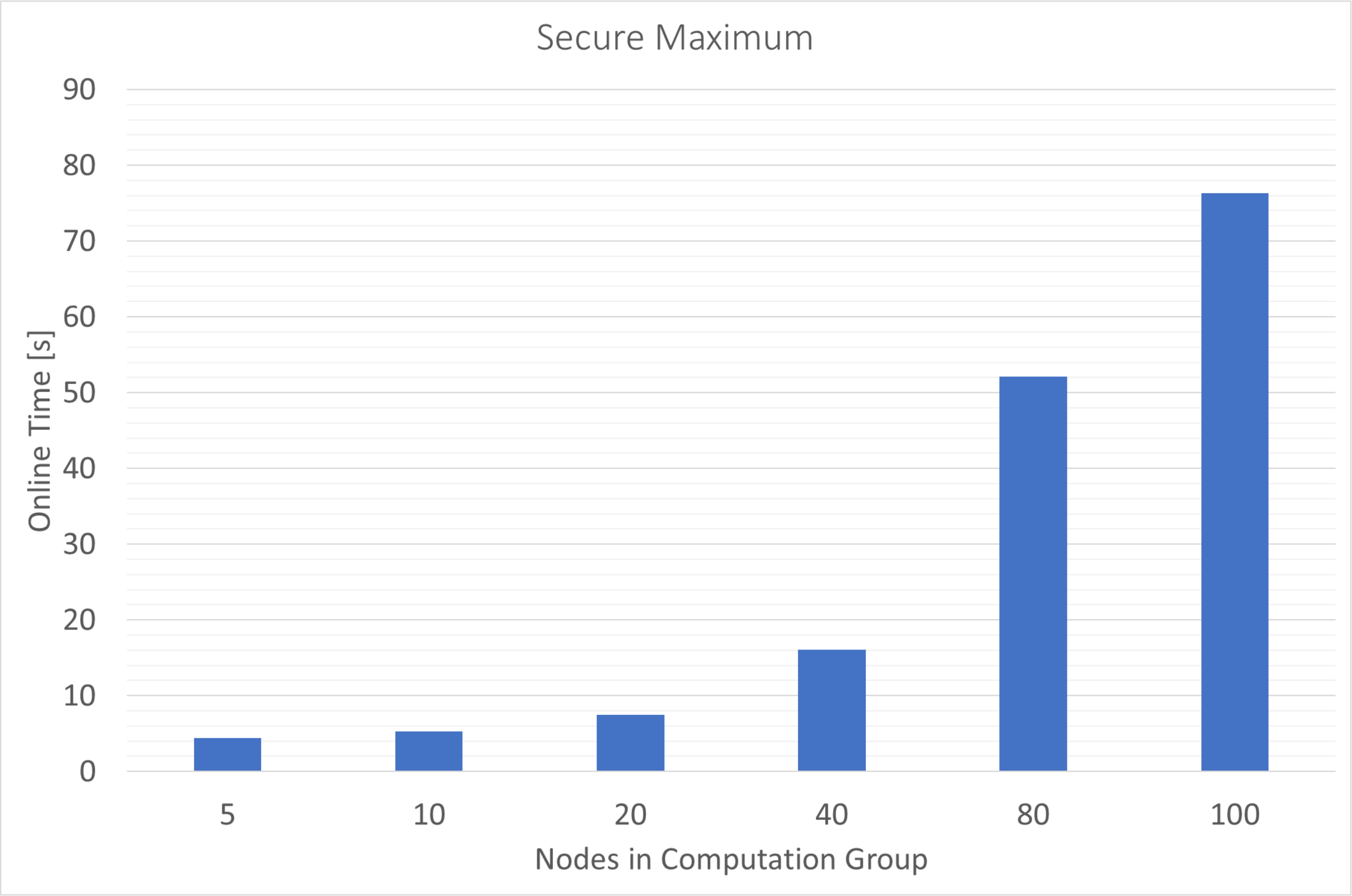
Discussion: Simulation Results

- Discovery bottle neck
- Computation fast



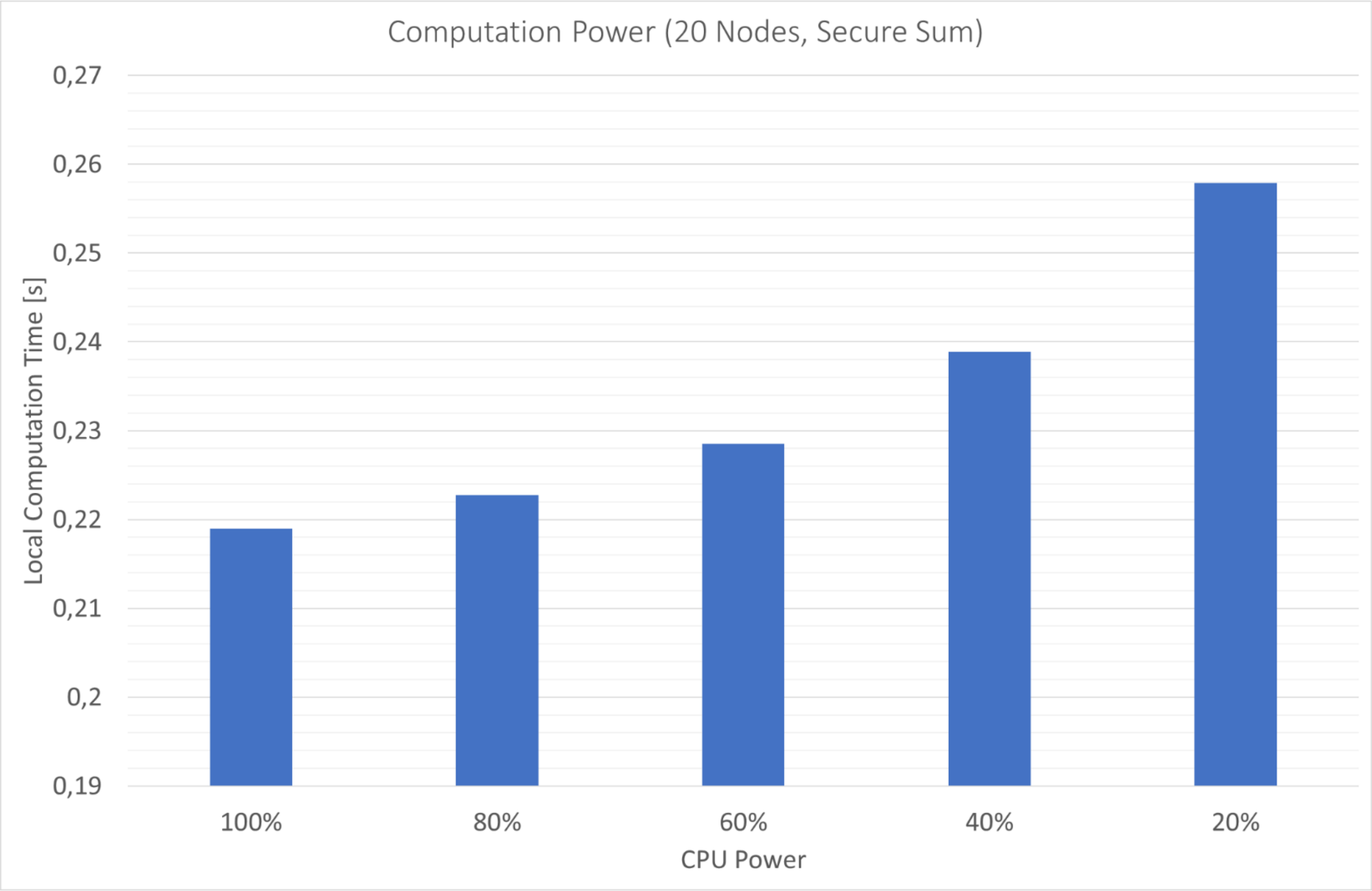
Discussion: Simulation Results

- Number of messages squared
- Larger computation groups: high risk of network partition



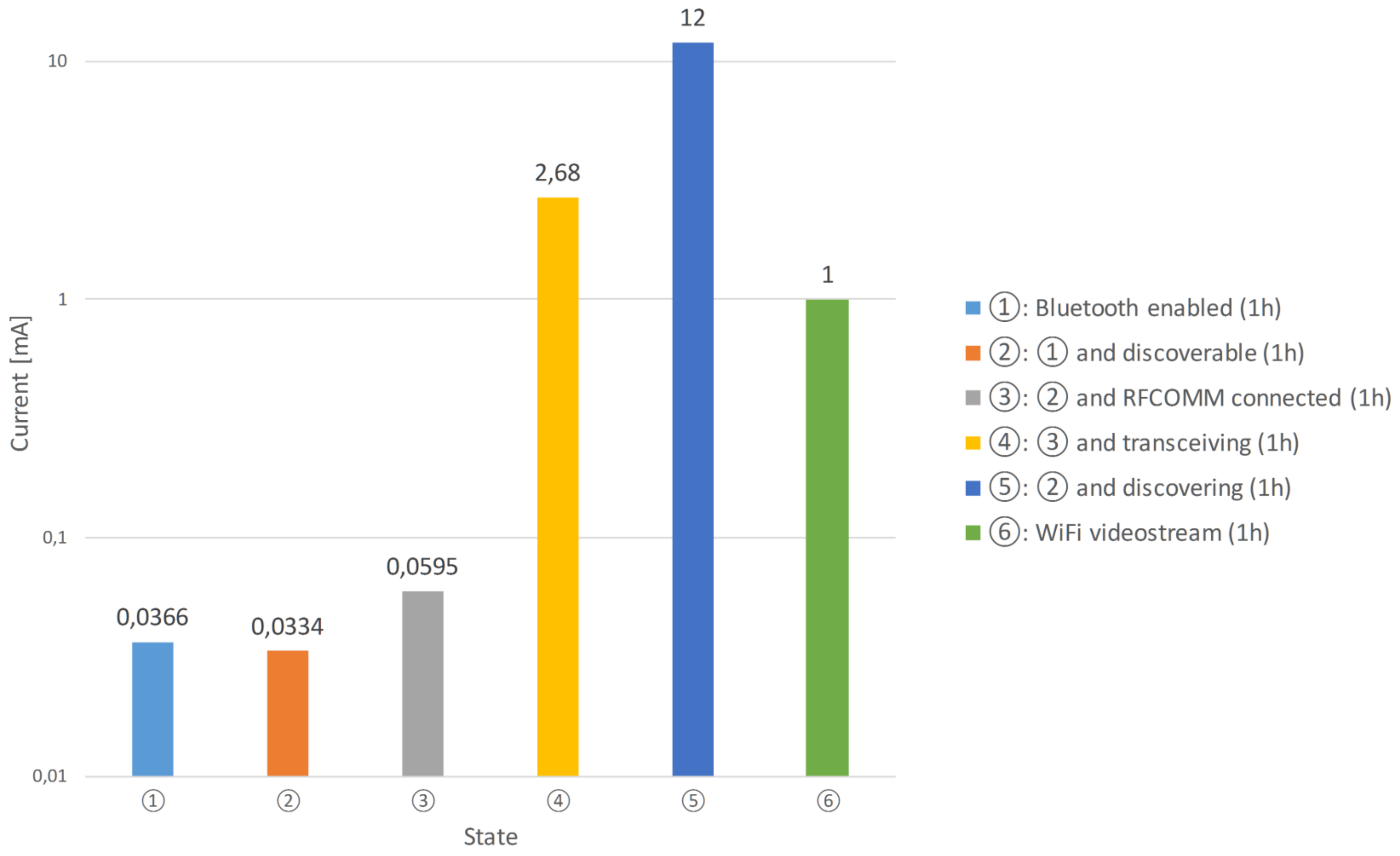
Discussion: Simulation Results

- Influence of computation power low compared to message overhead



Discussion: Android Results

- Discovery and transmission expensive
- But:
  - discovery << display
  - Discovery << capacity



Discussion: Problems

- Leftovers
- Strong coupling with Bluetooth
- Different behavior on different Android versions

Outlook

- Implementation of missing features
  - Distributed Database
  - Timeout Callbacks

Outlook: Improvements

- Separate MANET features into standalone library
- Extend with other wireless technologies
- Use BLE advertisement instead of Bluetooth Classic discovery
- Leftovers:
  - Minimum/Maximum: compute with previous results
  - Sum: TODO

Outlook: Usage

- Host as open source project
- Write paper on combination on secure multi-party gamification to gain attention
- Extend simulation script
- Implement into Hygiene Games for a study