Smart Contracts for Crowdsourced Spectrum Sensing

Aalto CS Forum Talk,

March 1st, 2019

Suzan Bayhan

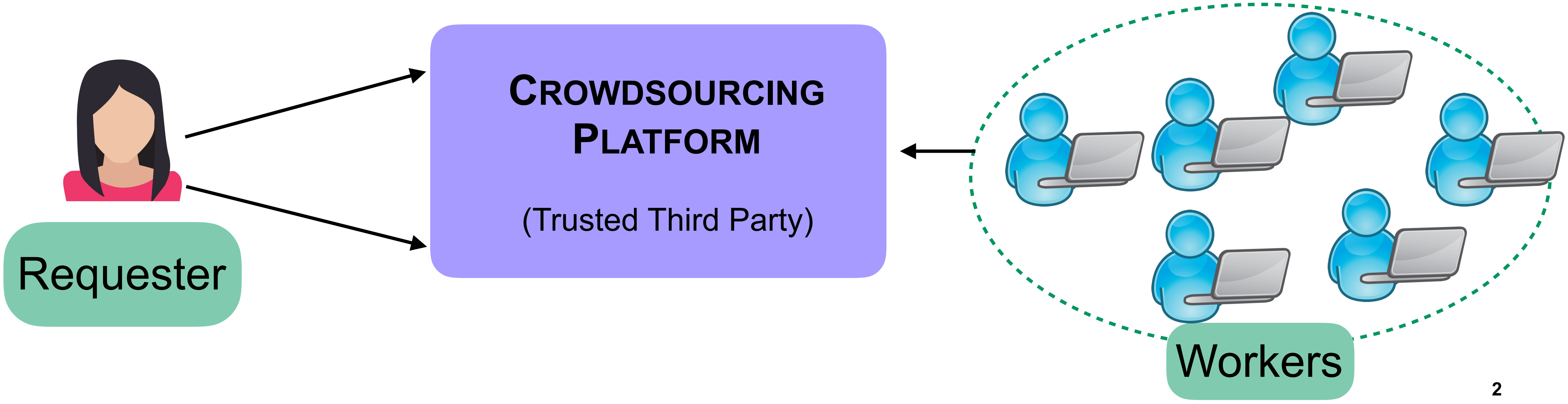
Telecommunication Networks Group, TU Berlin

suzanbayhan.github.io

Collaborators: Anatolij Zubow, Piotr Gawlowicz, Adam Wolisz

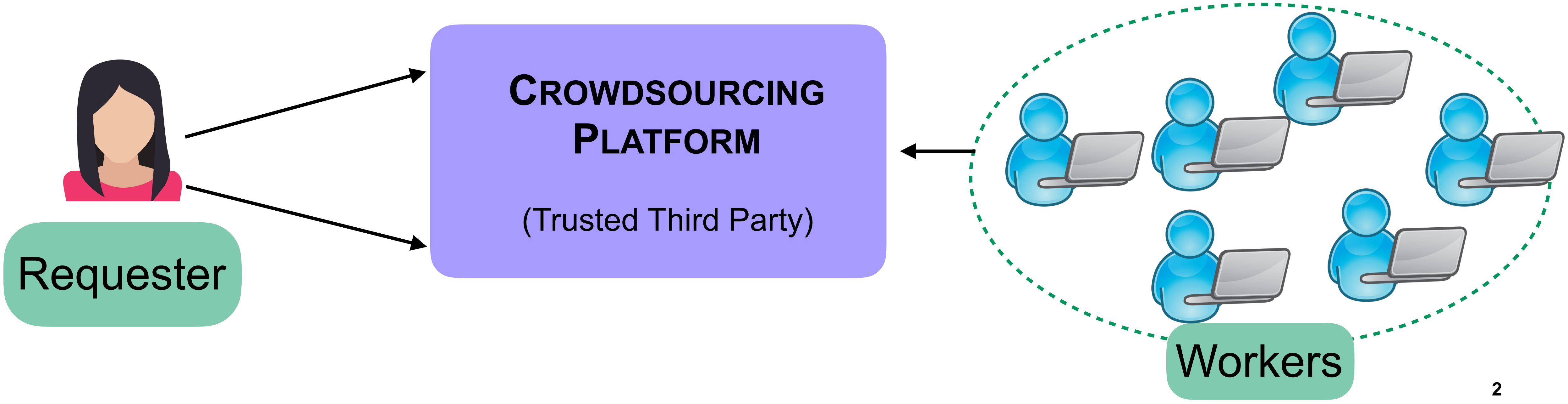
Technische Universität Berlin

Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

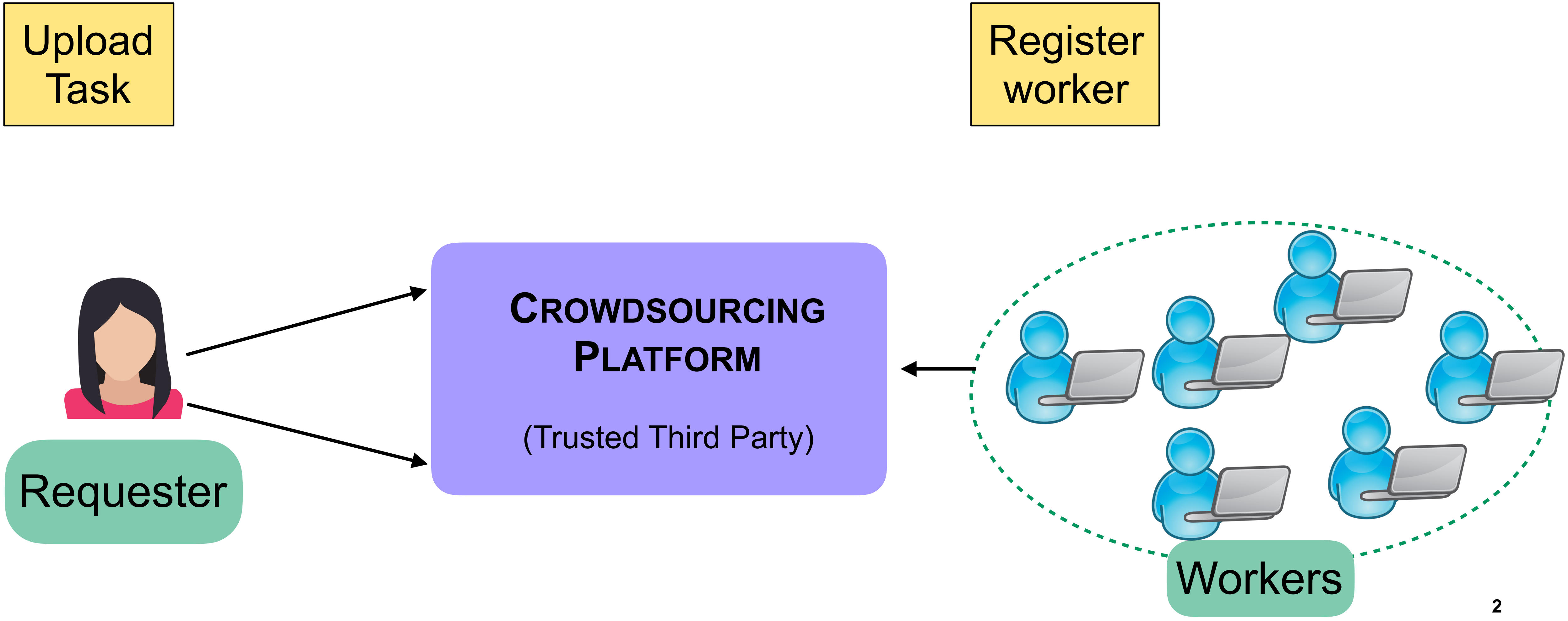


Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

Upload Task



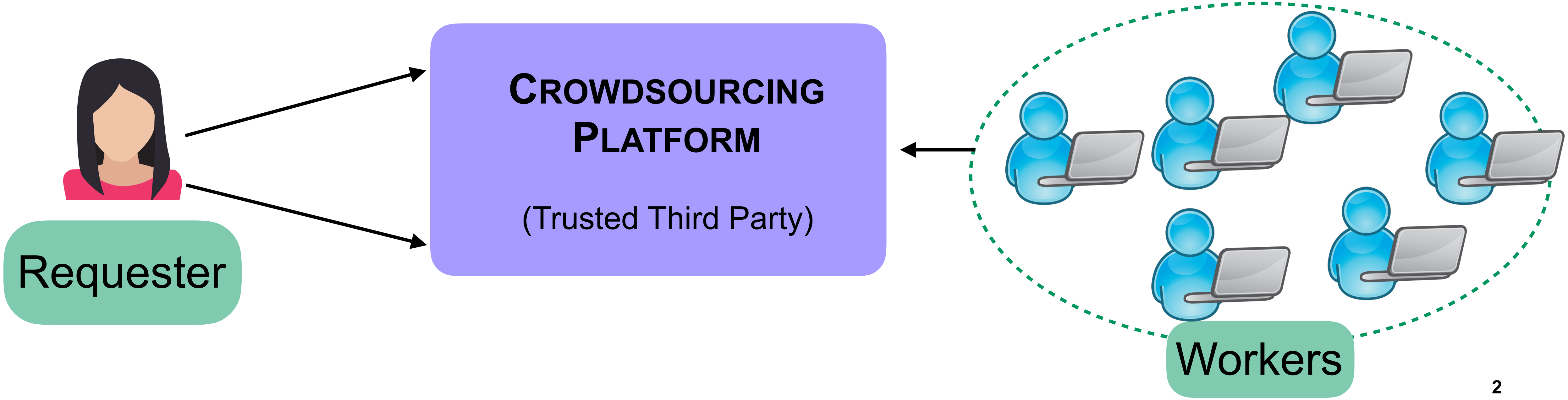
Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform



Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

Upload Task	Select Workers

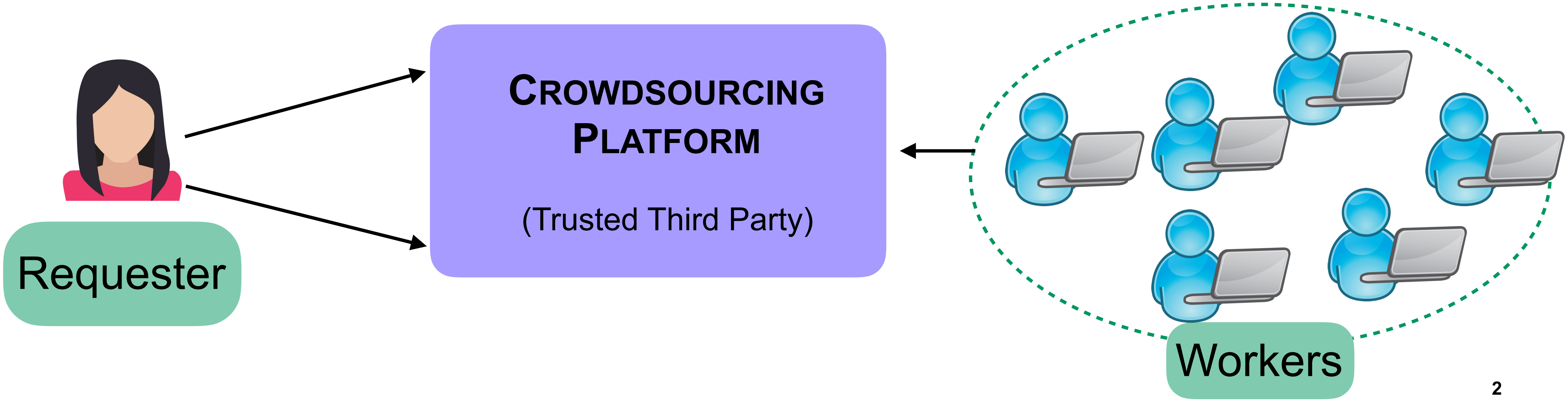
Register worker



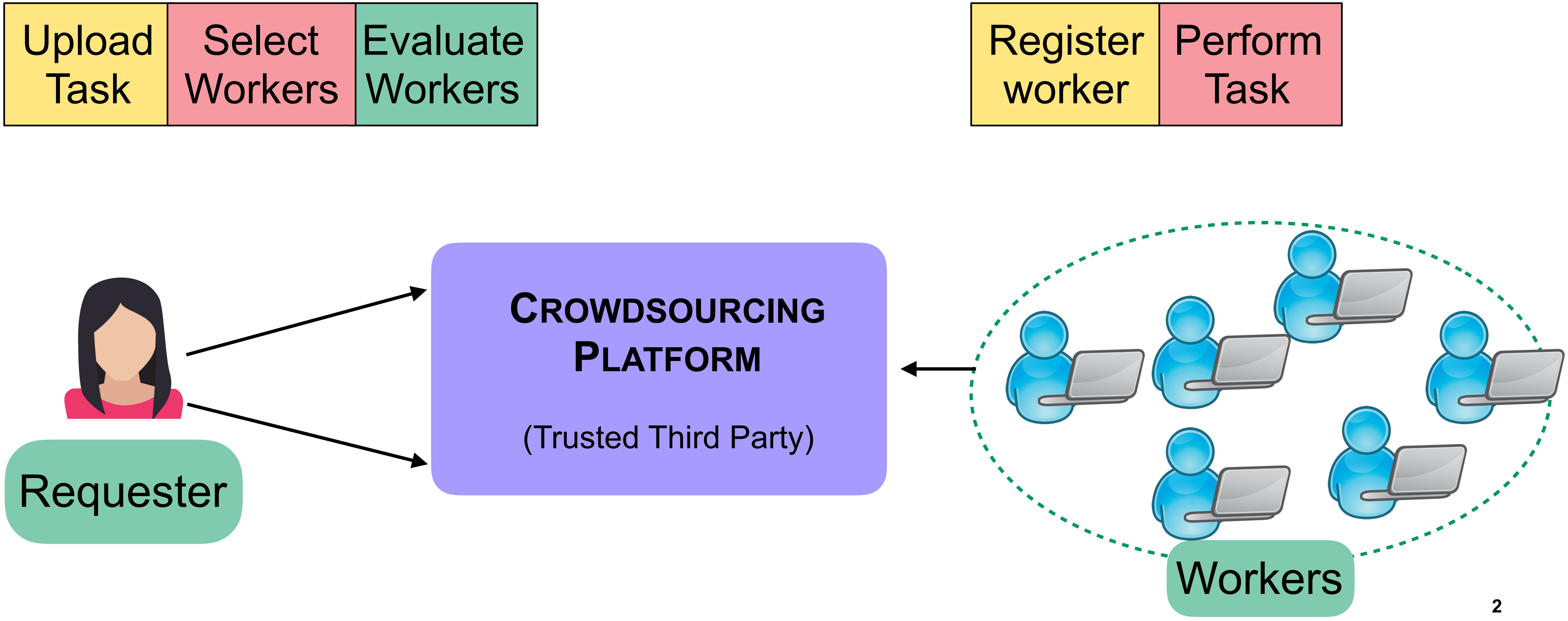
Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

Upload Task	Select Workers

Register worker	Perform Task



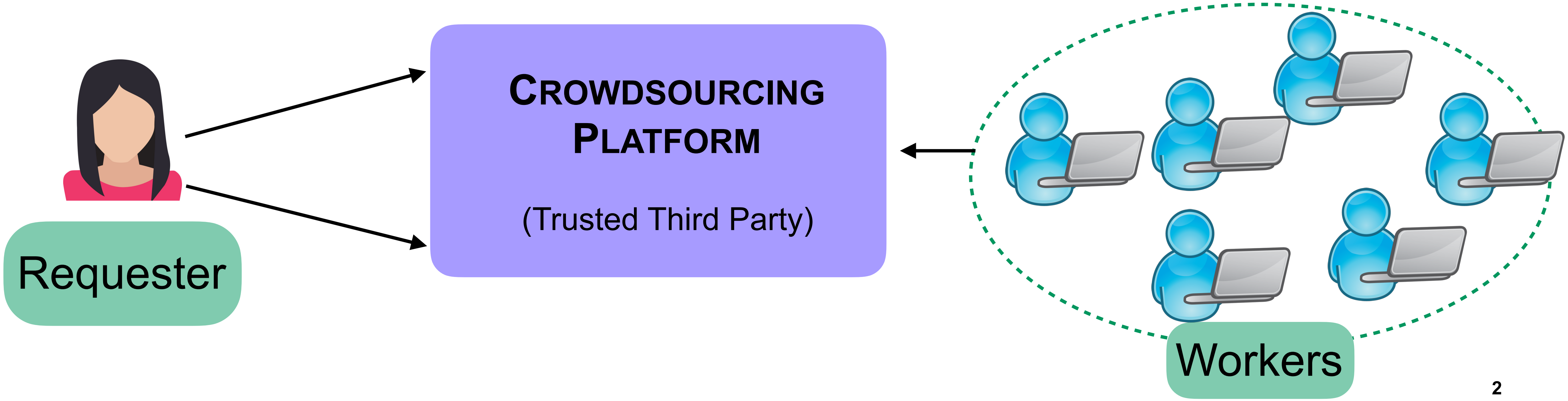
Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform



Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

Upload Task	Select Workers	Evaluate Workers	Pay Workers

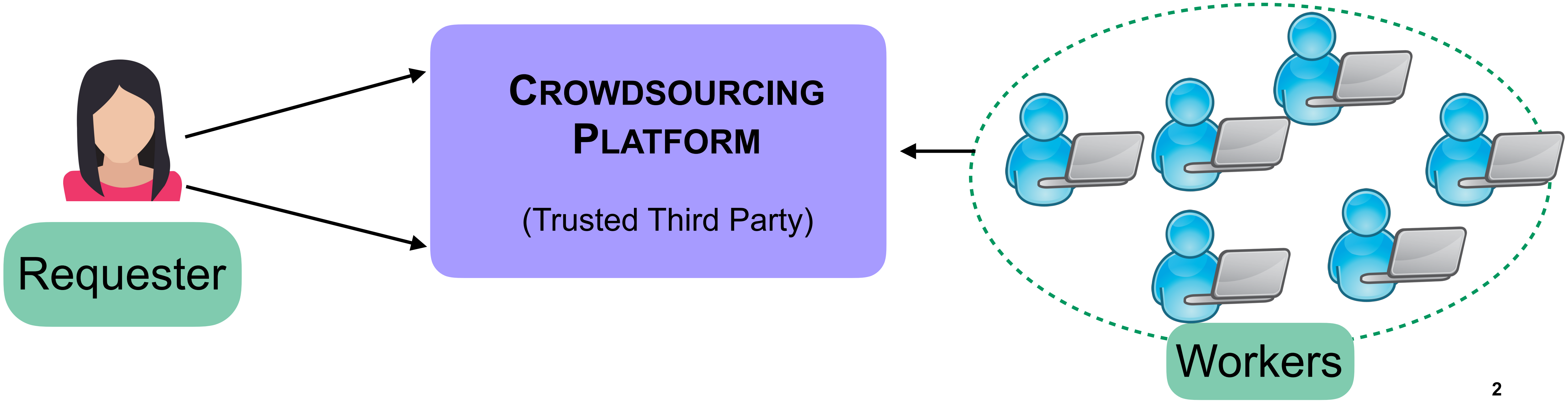
Register worker	Perform Task



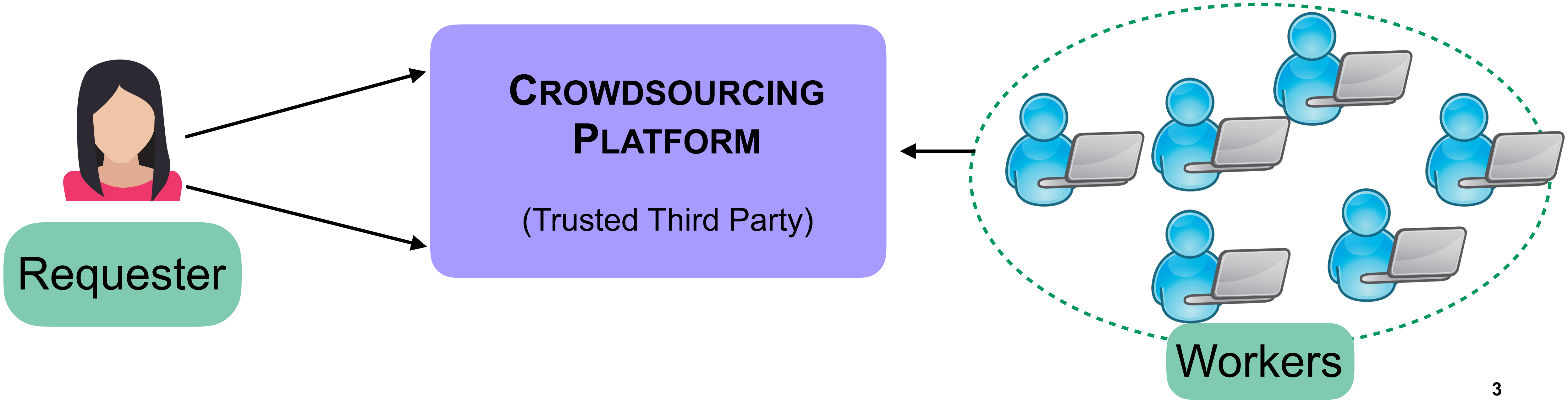
Typical centralized crowdsourcing solutions: requester, workers, crowdsourcing platform

Upload Task	Select Workers	Evaluate Workers	Pay Workers

Register worker	Perform Task	Evaluate Requester

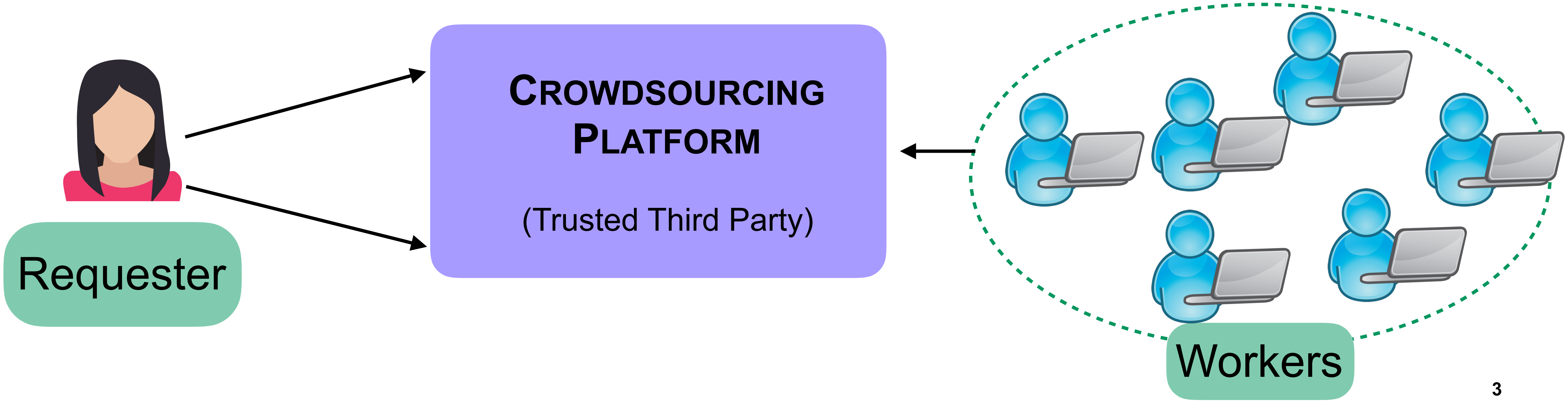


But, several shortcomings



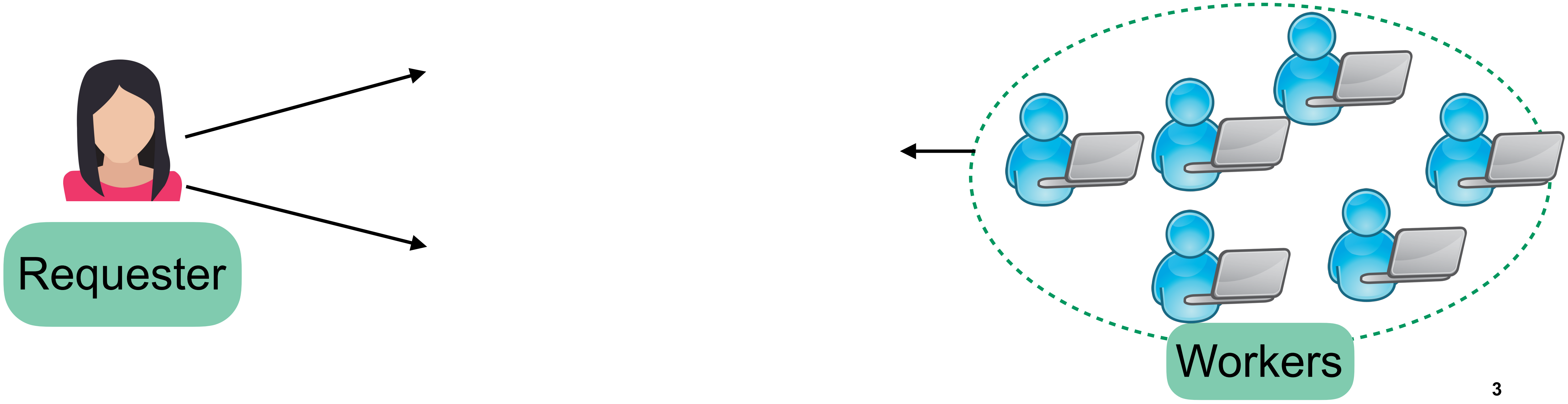
But, several shortcomings

- Trust
- Service fees
- Single point of failure
- Dispute resolving (repudiation of payments)
- Prone to attacks or fraud (free-riders)
- Heterogeneity (e.g., uncalibrated, low-precision sensors)



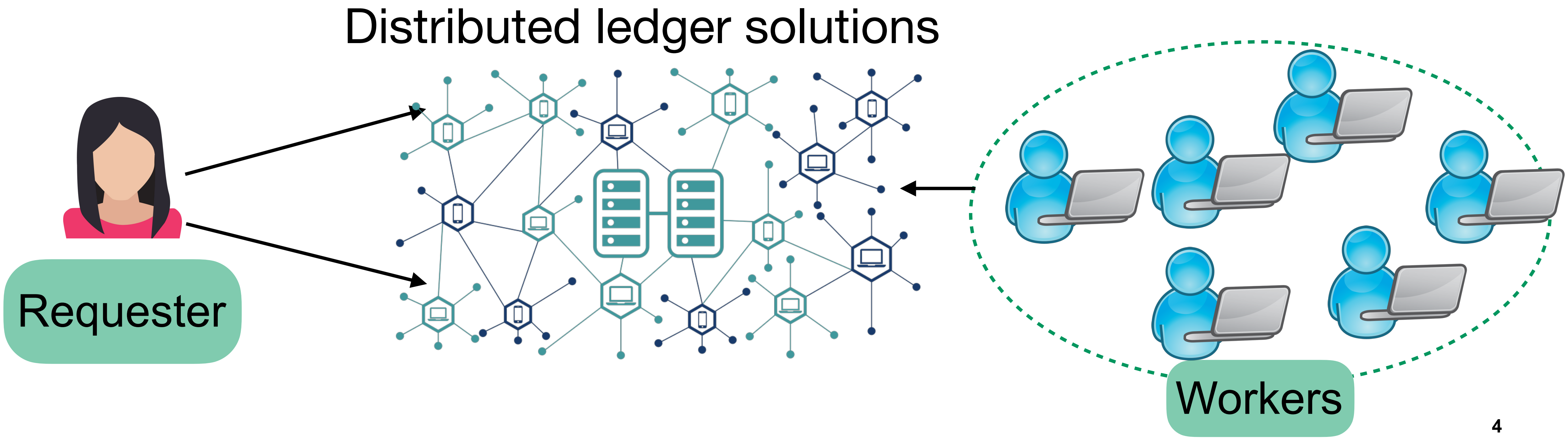
But, several shortcomings

- Trust
- Service fees
- Single point of failure
- Dispute resolving (repudiation of payments)
- Prone to attacks or fraud (free-riders)
- Heterogeneity (e.g., uncalibrated, low-precision sensors)



But, several shortcomings

- Trust
- Service fees
- Single point of failure
- Dispute resolving (repudiation of payments)
- Prone to attacks or fraud (free-riders)
- Heterogeneity (e.g., uncalibrated, low-precision sensors)

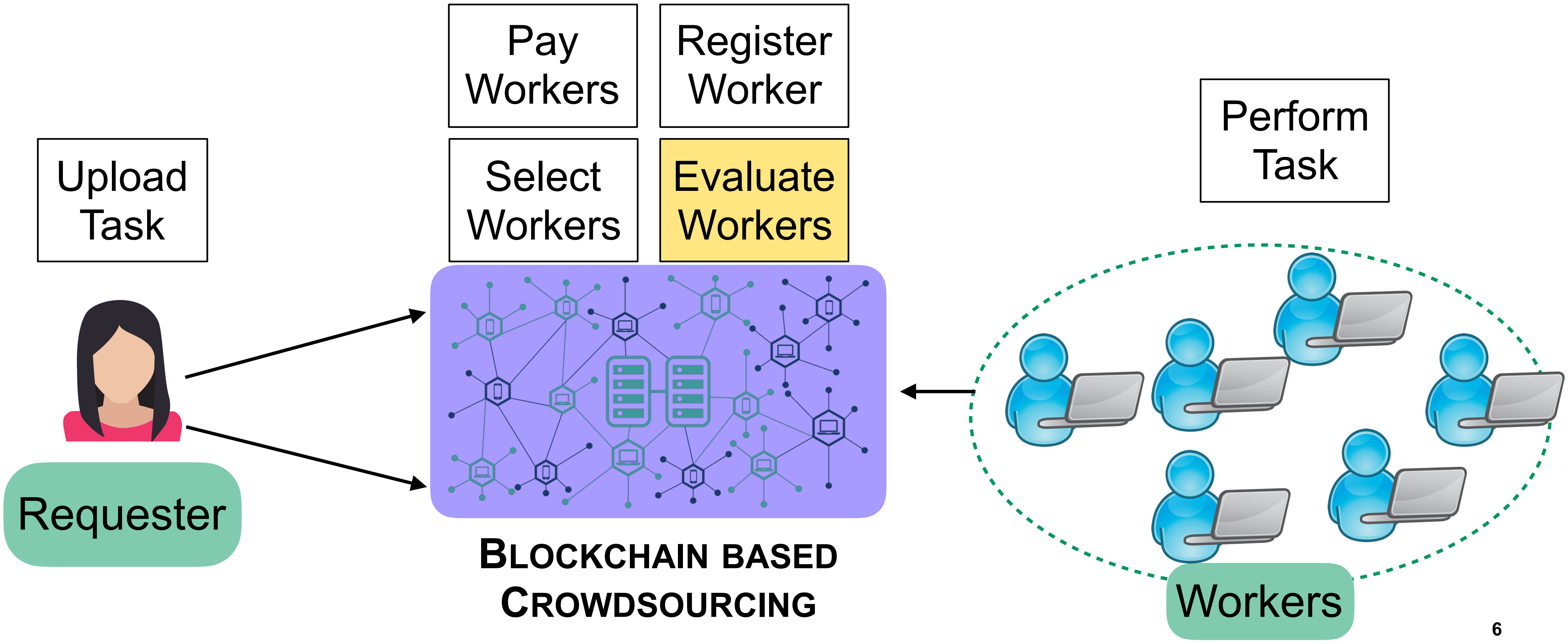


Distributed Ledgers for Crowdsourcing

- a decentralized crowdsourcing system
- smart contracts (SC) to implement functions of a crowdsourcing platform
- requires a secure environment which is decentralized, unalterable and programmable: provided by Blockchain networks
- *transaction fee* for miners who confirm the transaction and support blockchain running persistently [Li 2019]

- Lu, Yuan, Qiang Tang, and Guiling Wang. "Zebralancer: Private and anonymous crowdsourcing system atop open blockchain." *IEEE ICDCS* 2018.
- M. Li *et al*., "CrowdBC: A Blockchain-based Decentralized Framework for Crowdsourcing," in *IEEE Transactions on Parallel and Distributed Systems*, 2019.

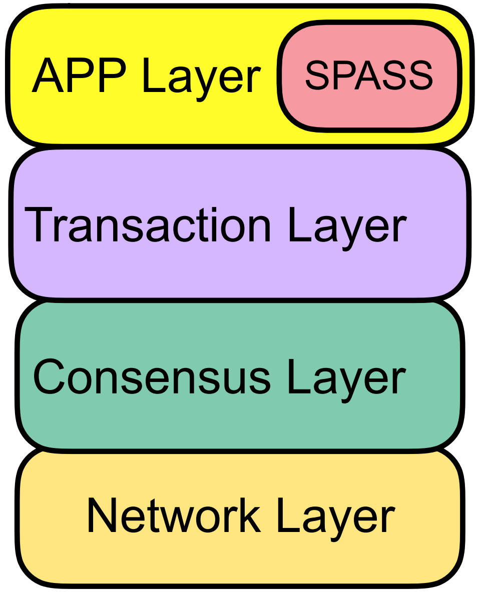
Challenge: SCs storage and processing limited



Spass: Spectrum Sensing as a Service via Smart Contracts

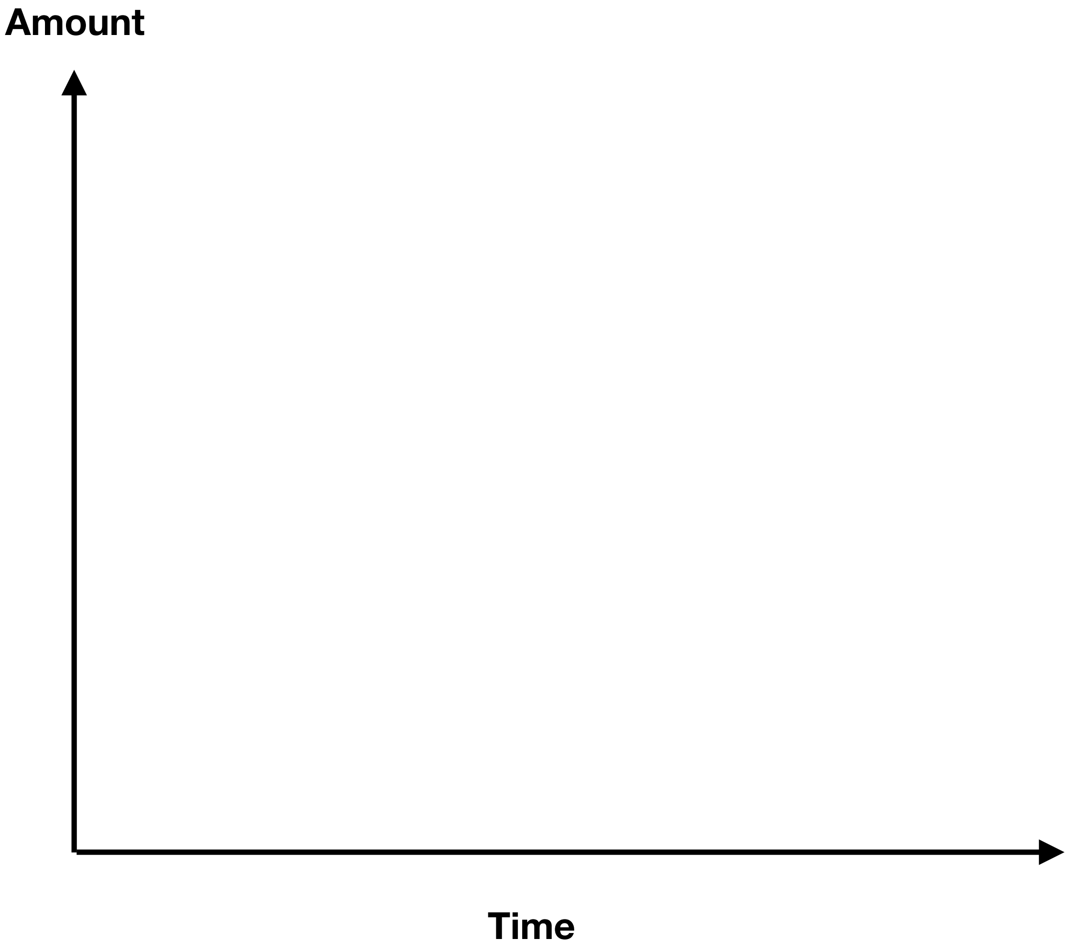
presented at IEEE DySPAN 2018

(under review, IEEE Transactions on Cognitive Communications and Networking)

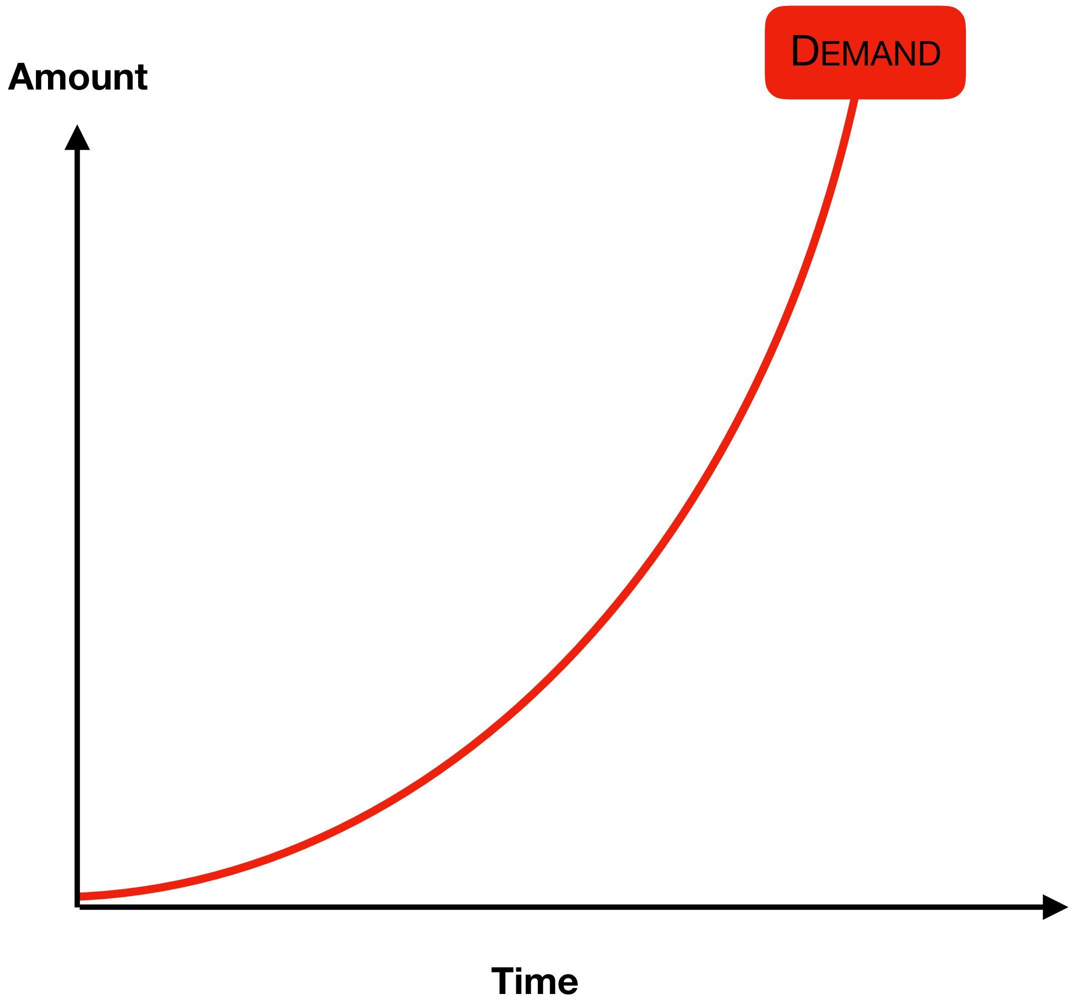


Need for more network capacity

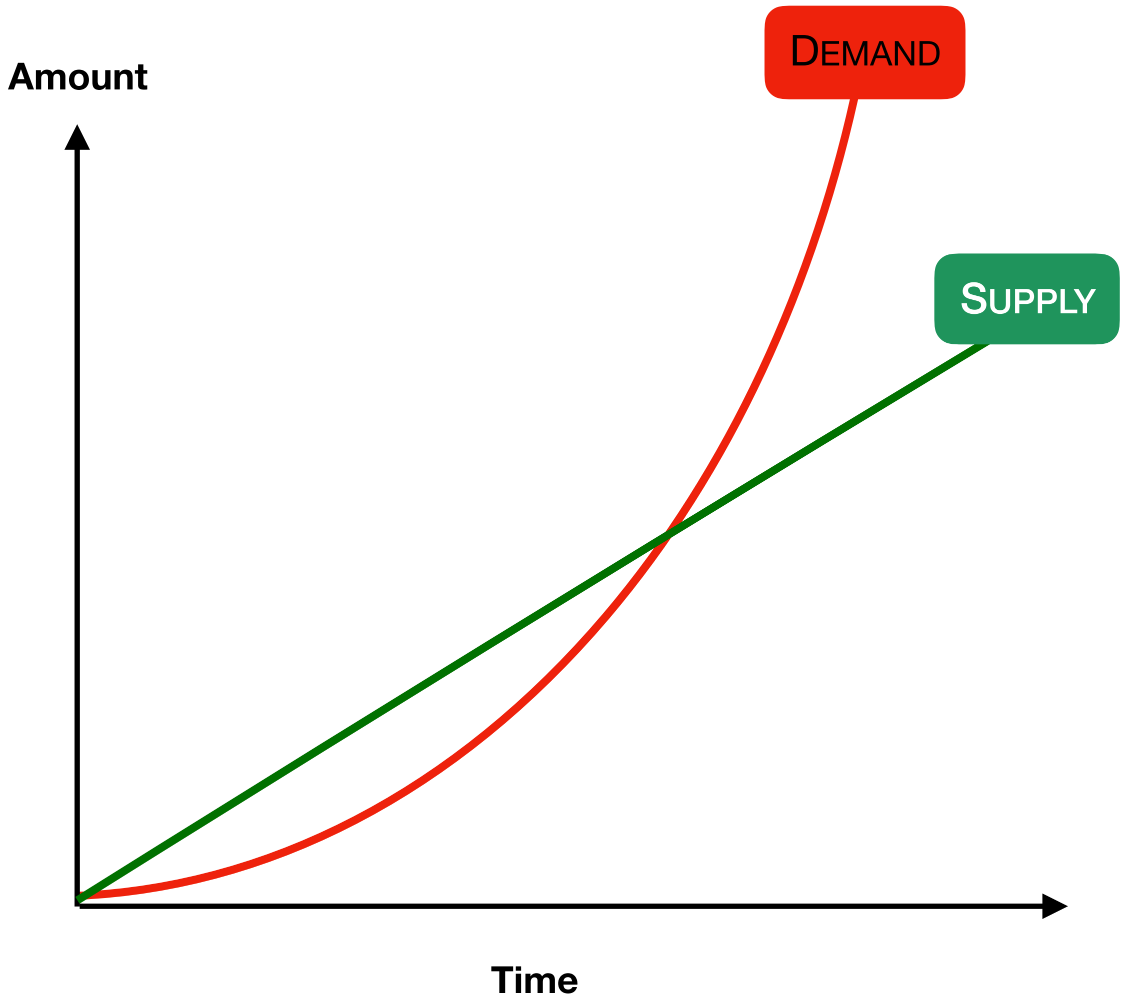
Need for more network capacity



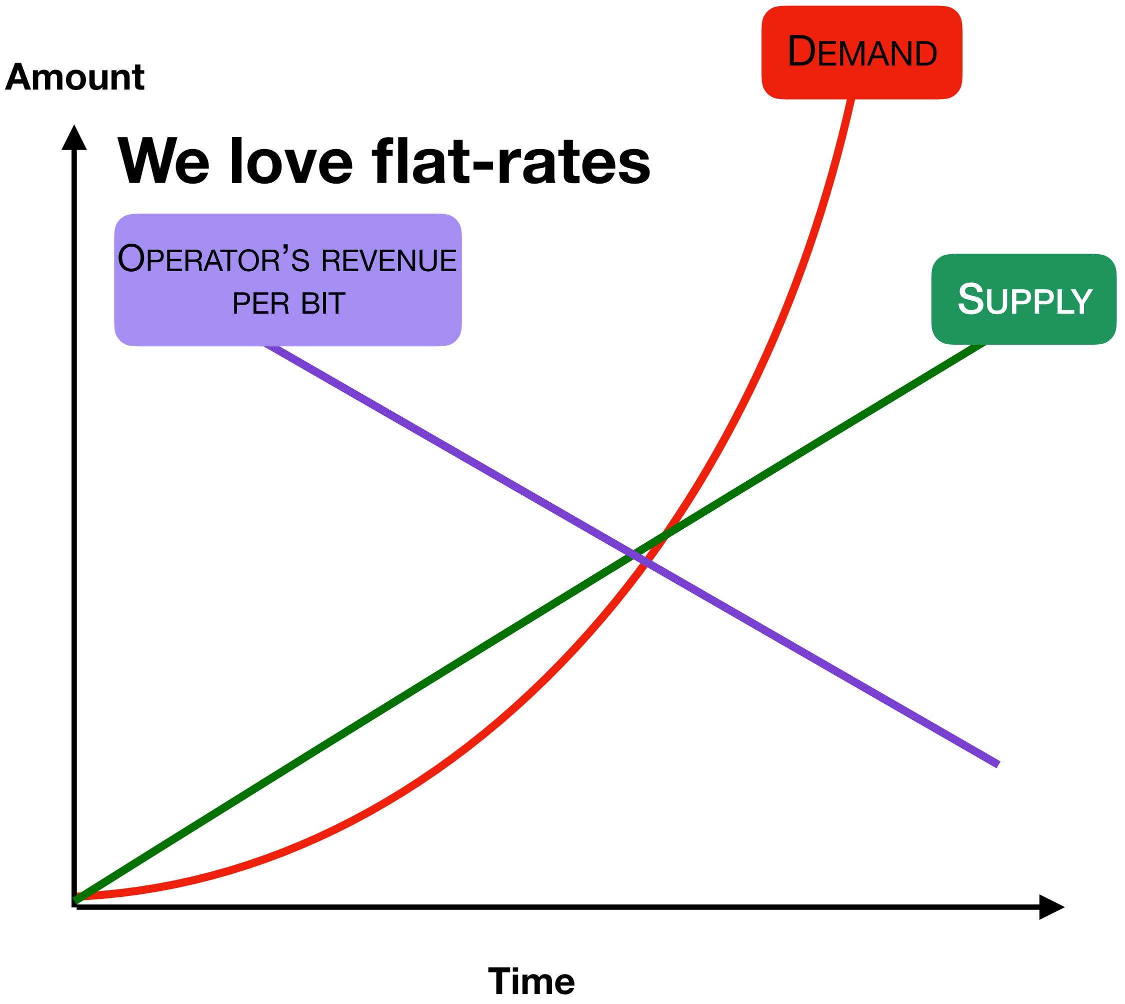
Need for more network capacity



Need for more network capacity

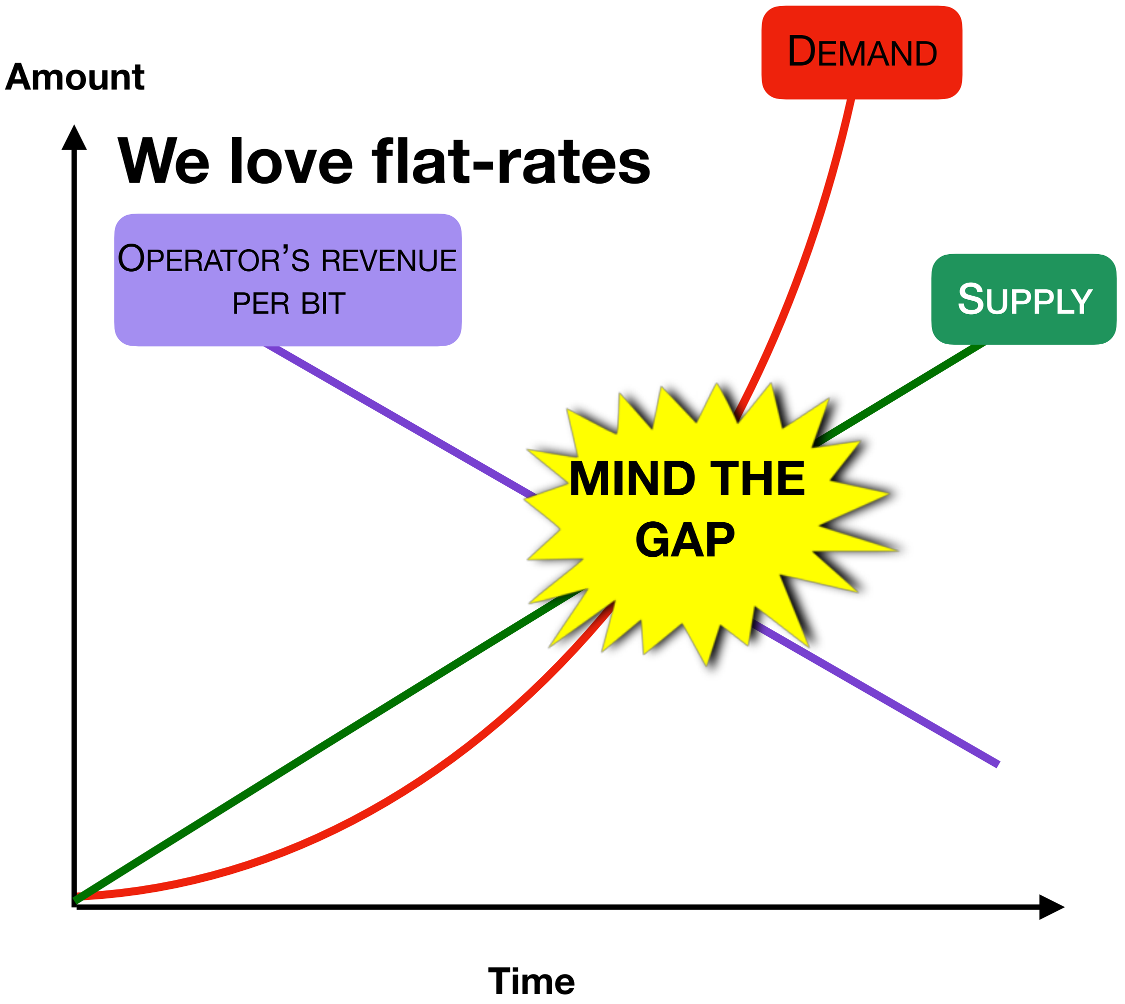


Need for more network capacity



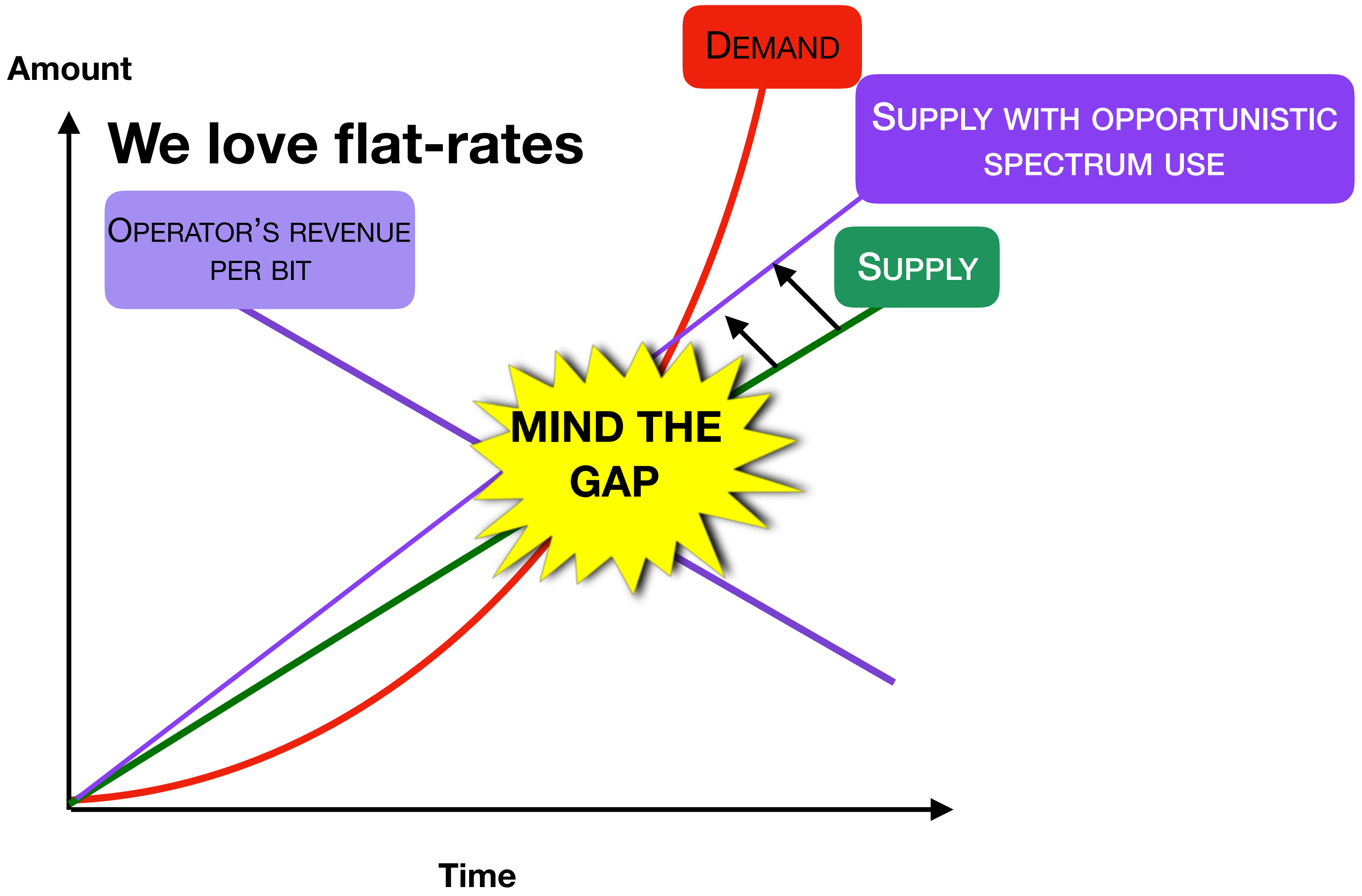
Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

Need for more network capacity



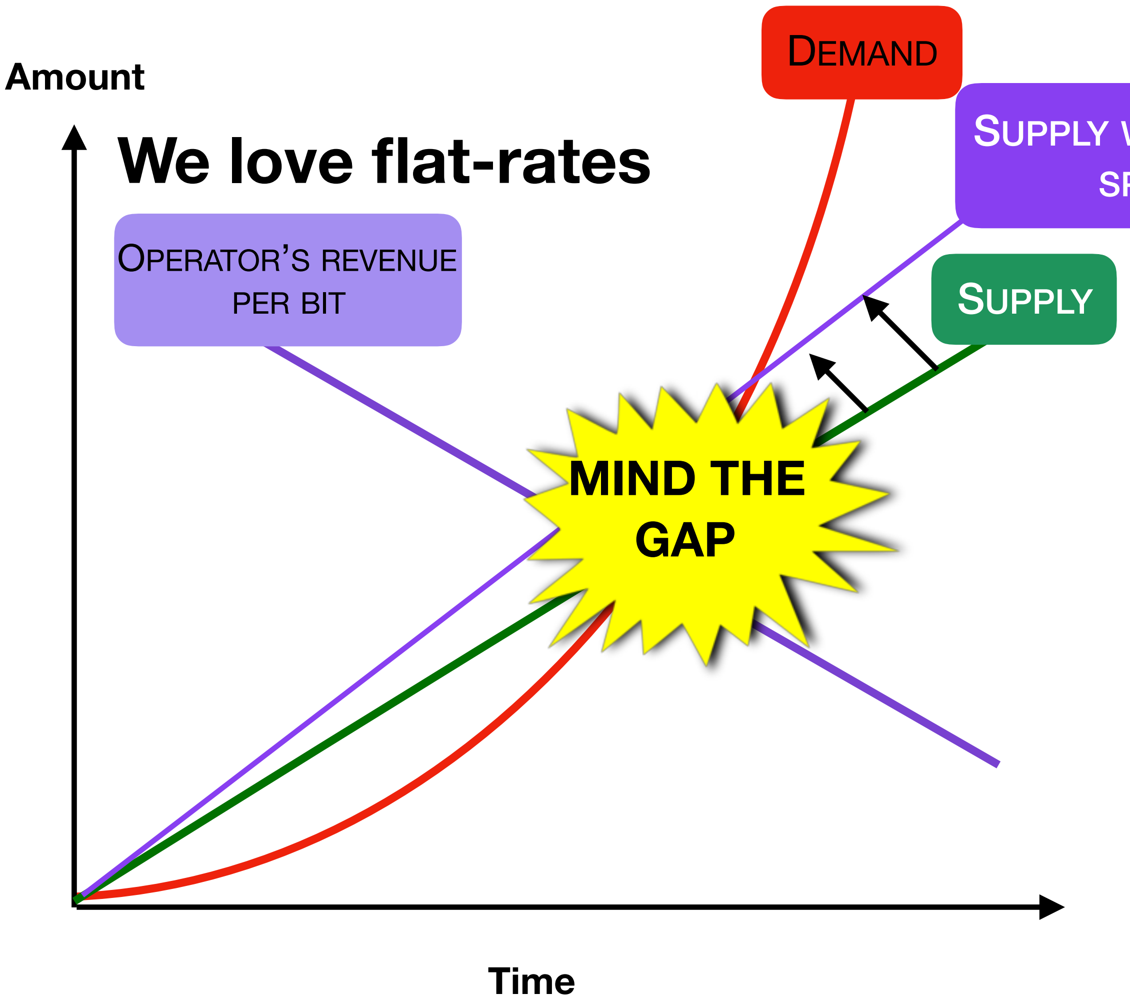
Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

Need for more network capacity

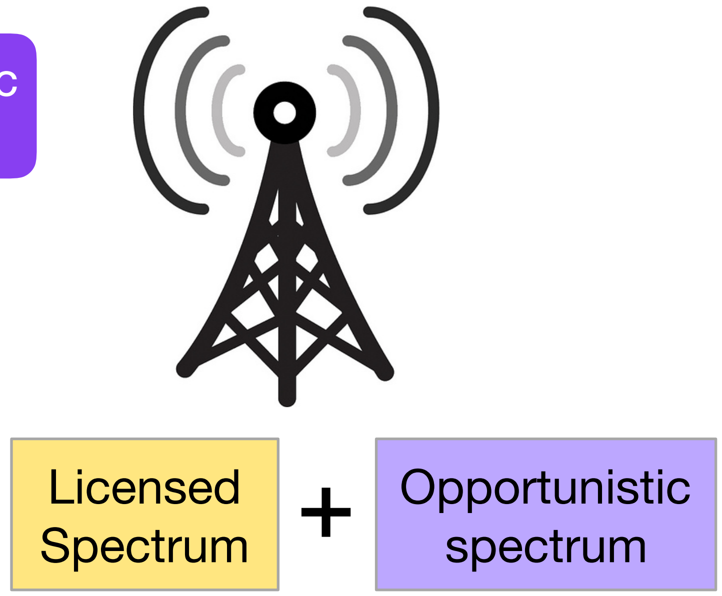


Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

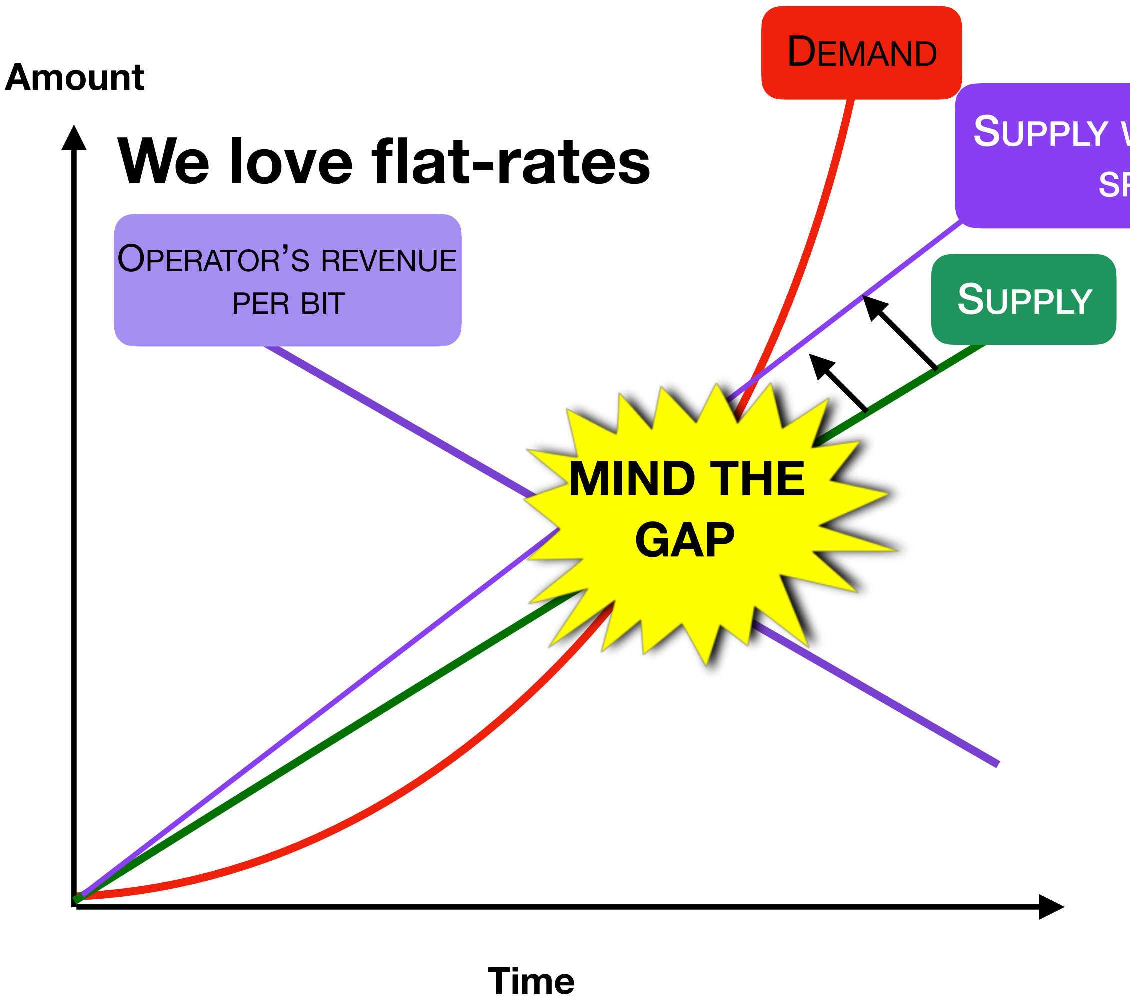
Need for more network capacity

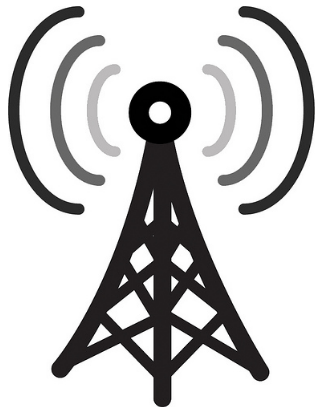


Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens



Need for more network capacity

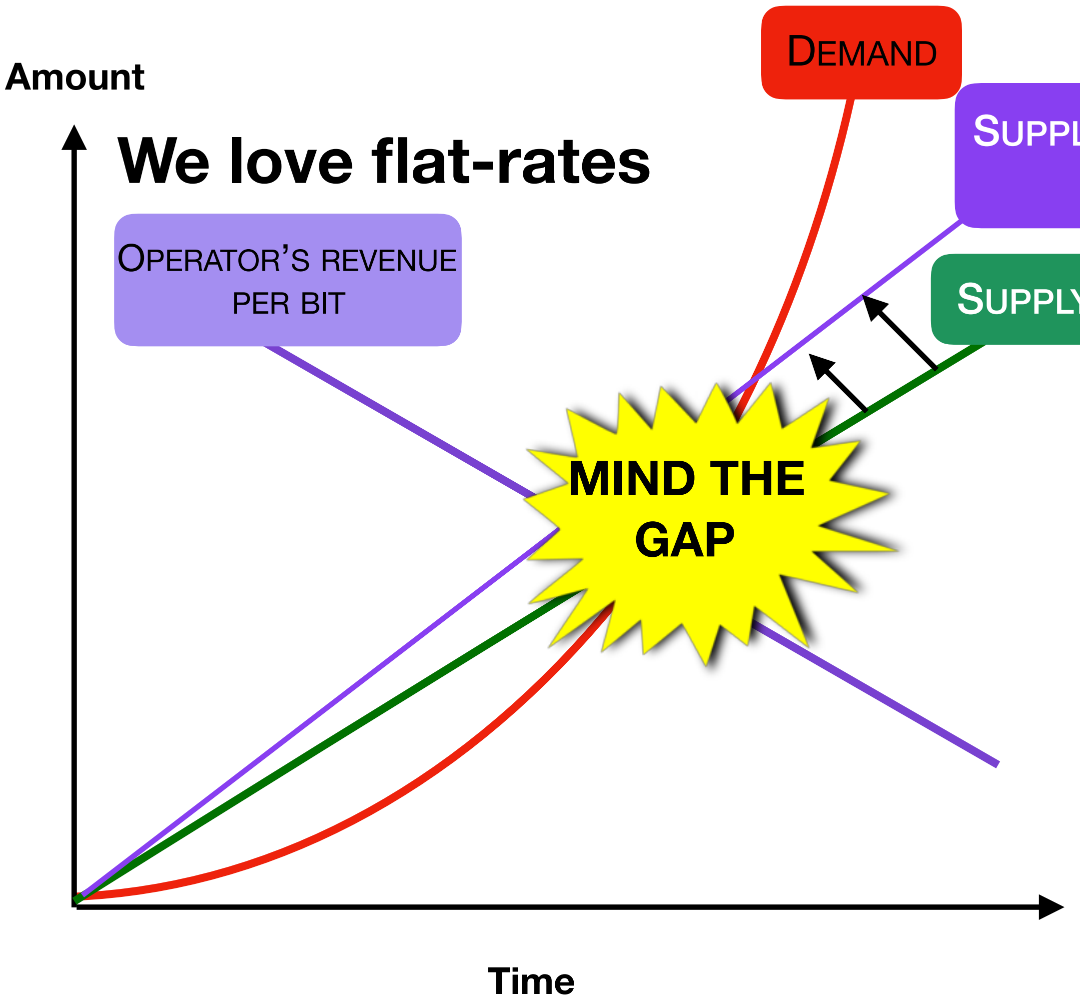


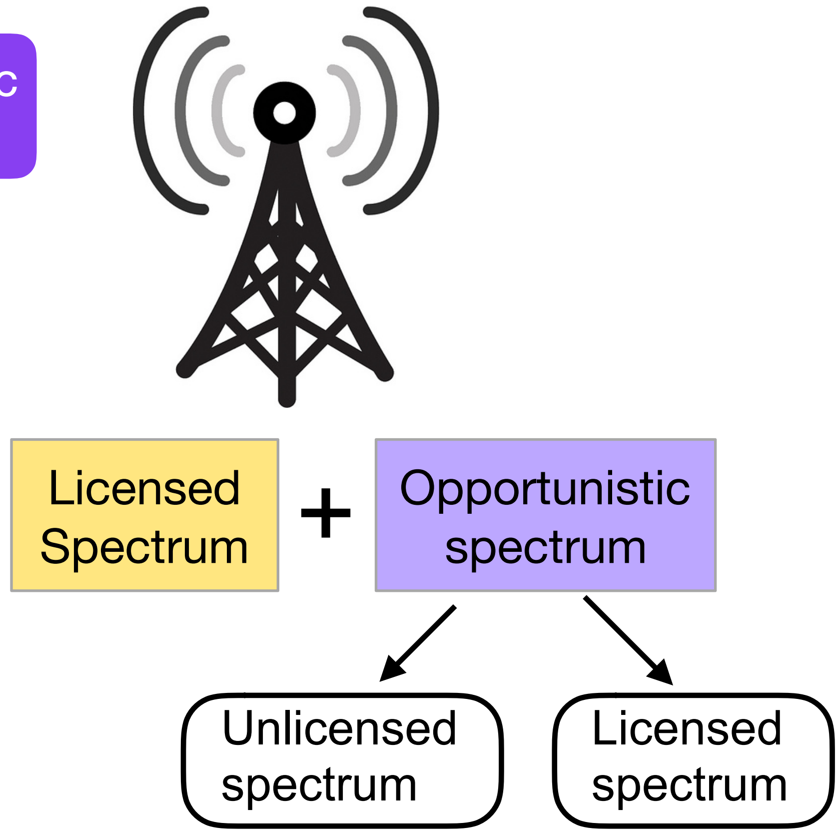


Licensed Spectrum + Opportunistic spectrum

Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

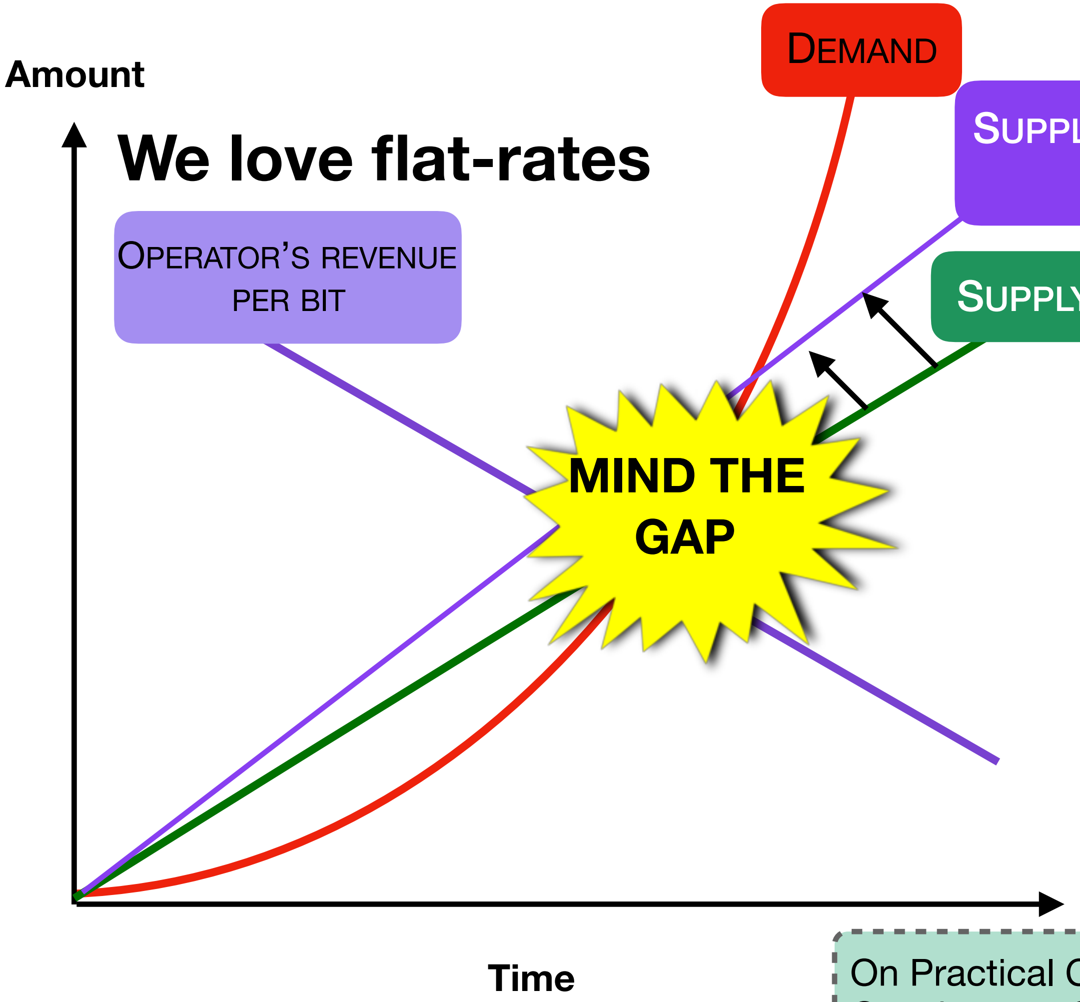
Need for more network capacity



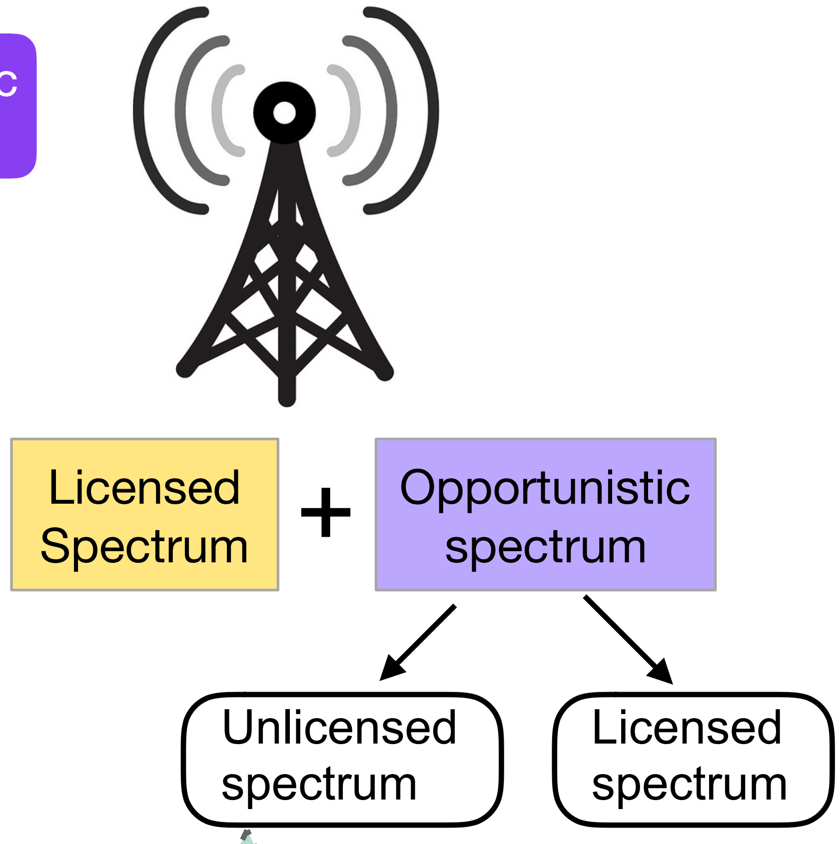


Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

Need for more network capacity

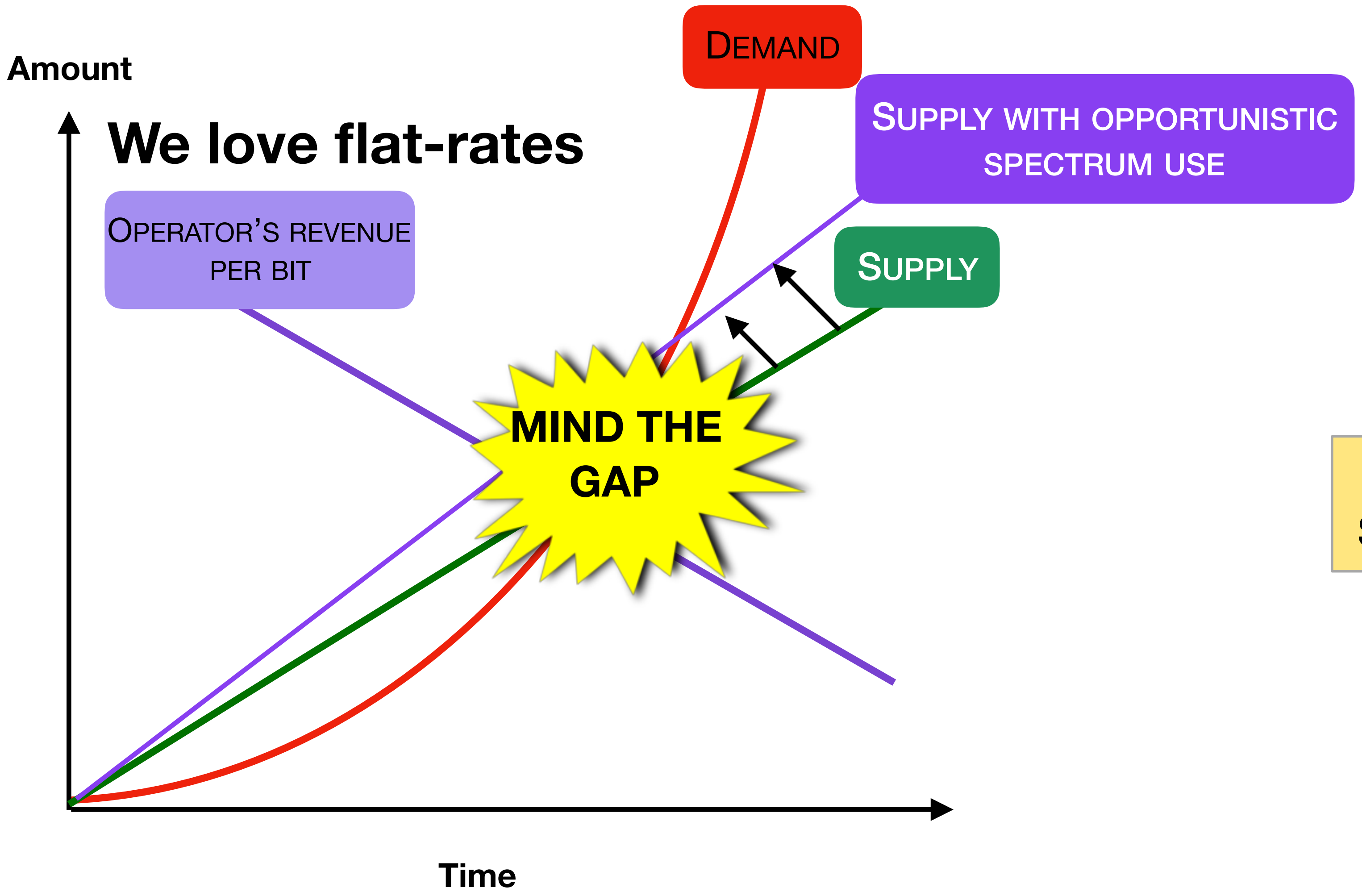


Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

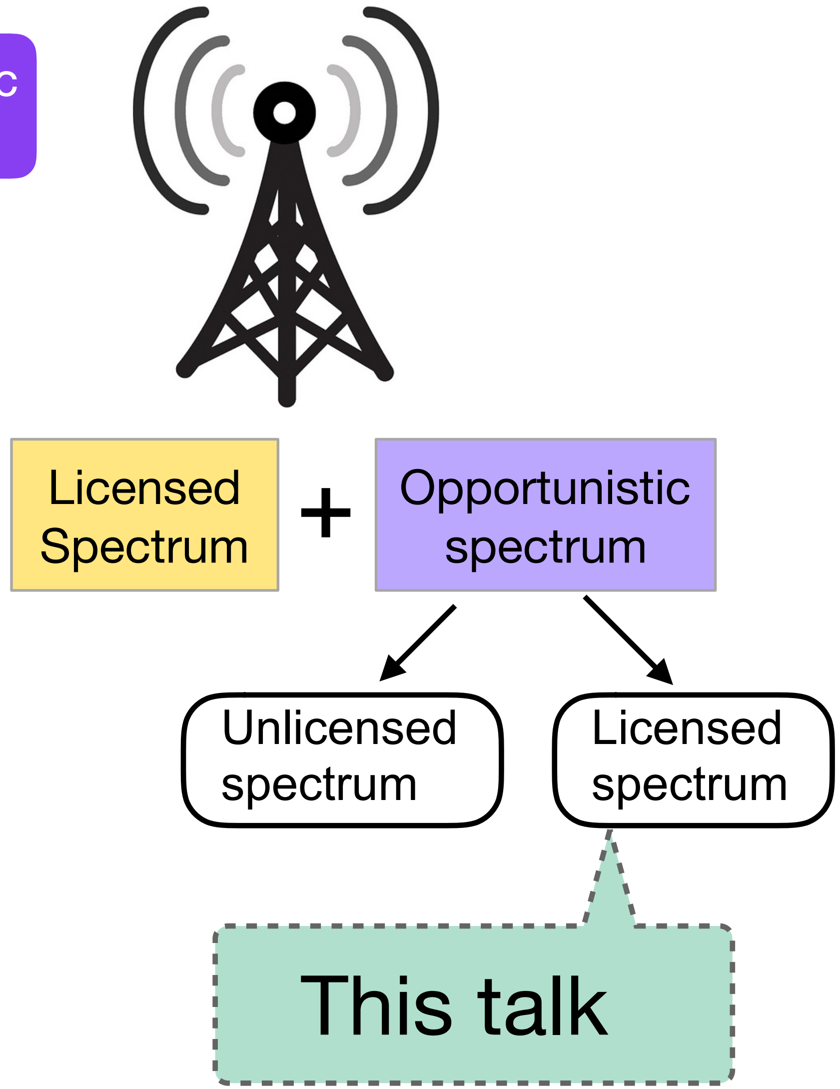


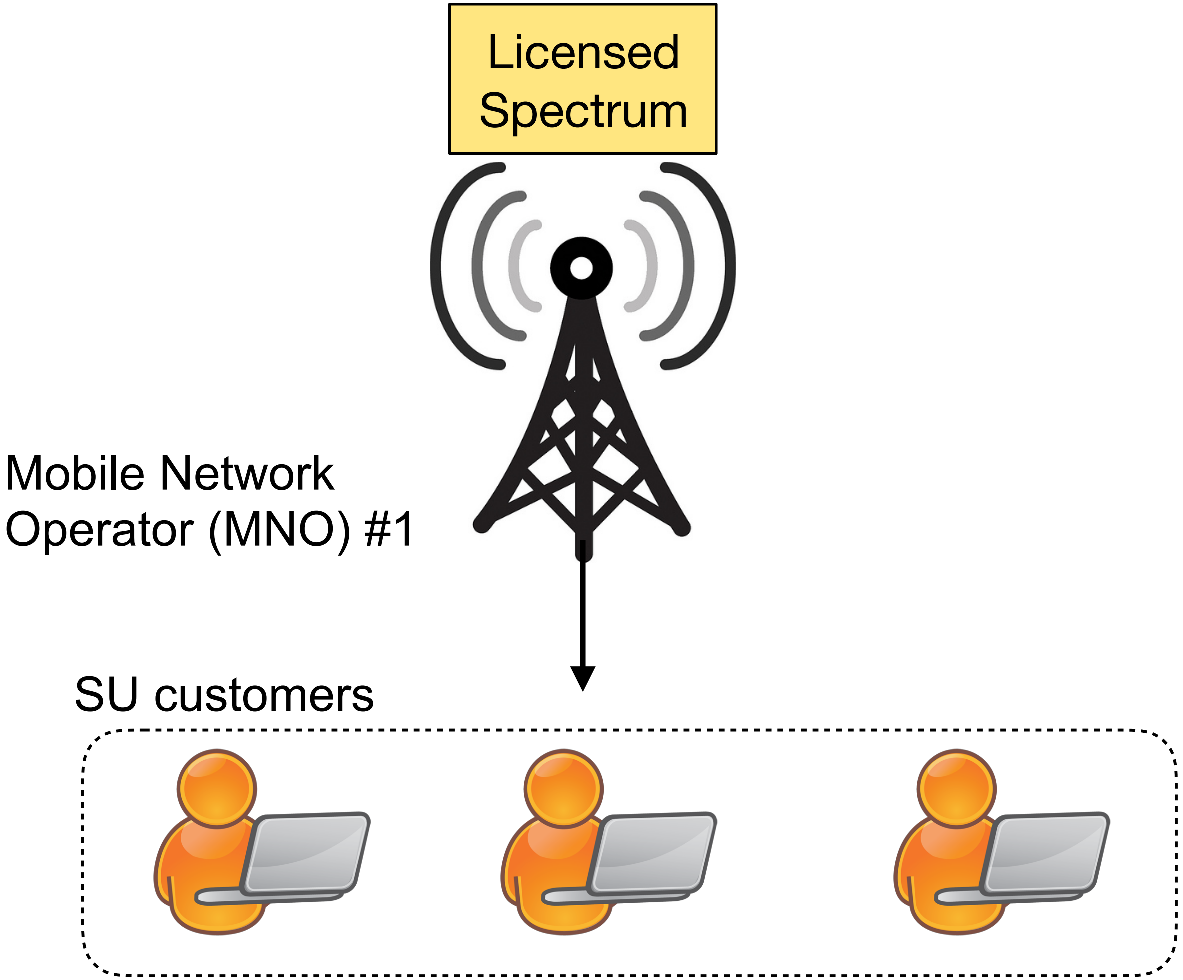
On Practical Coexistence Gaps in Space for LTE-U/WiFi Coexistence, European Wireless 2018.
Coexistence Gaps in Space: Cross-Technology Interference-Nulling for Improving LTE-U/WiFi Coexistence, IEEE WoWMoM 2018.

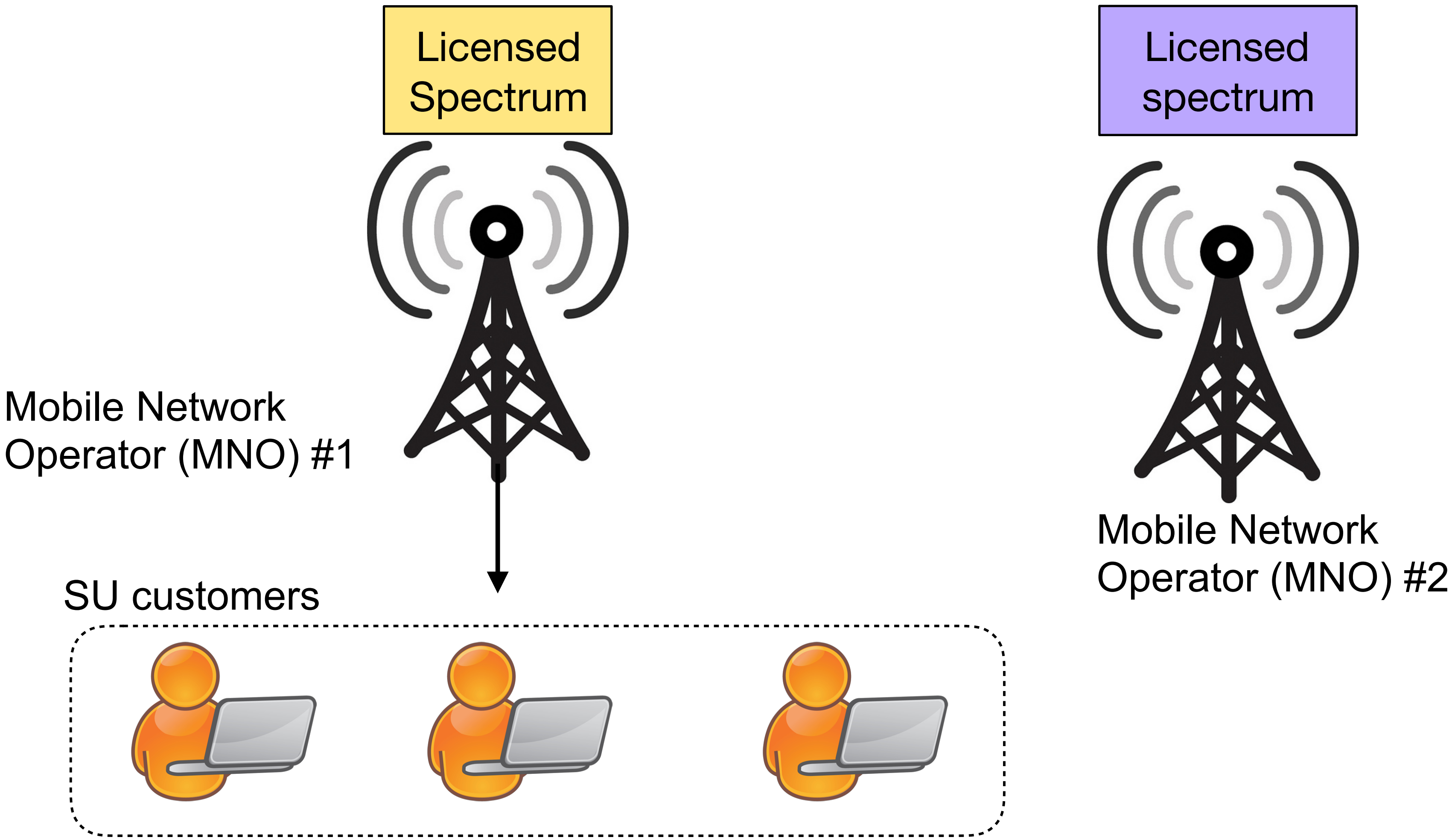
Need for more network capacity

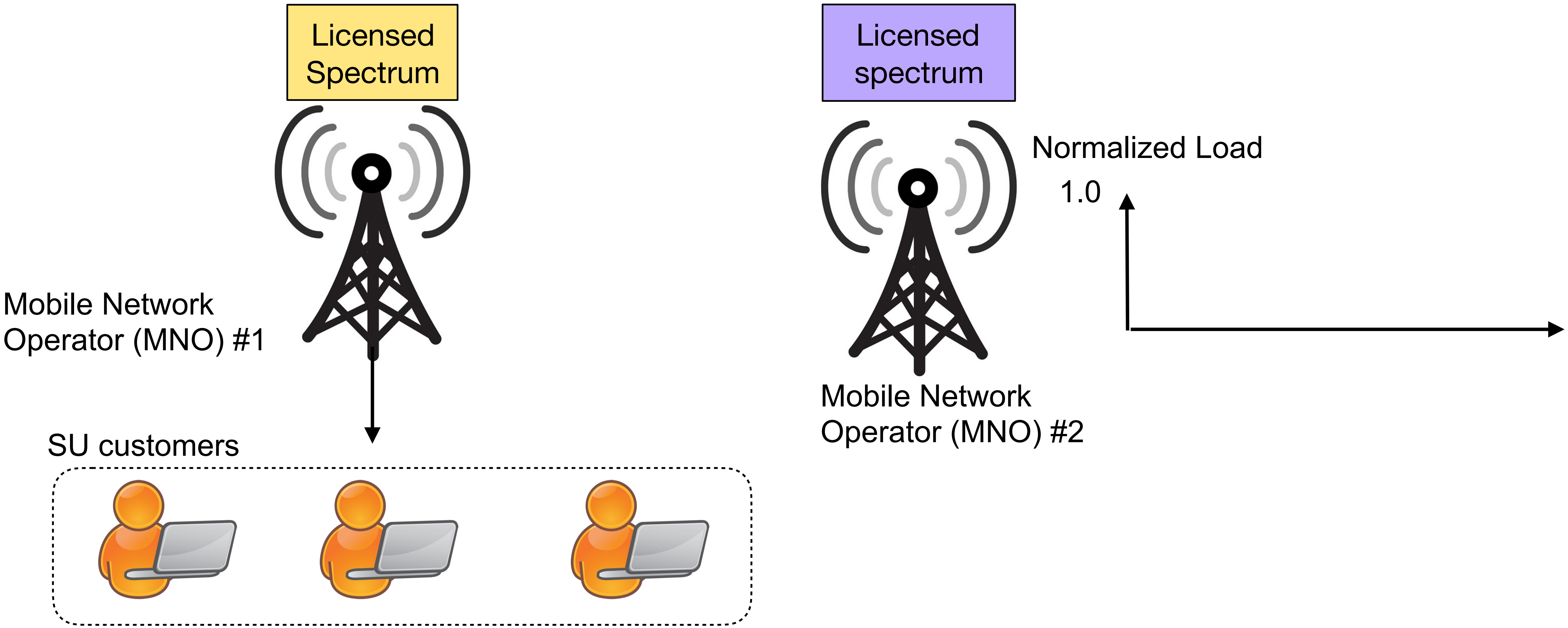


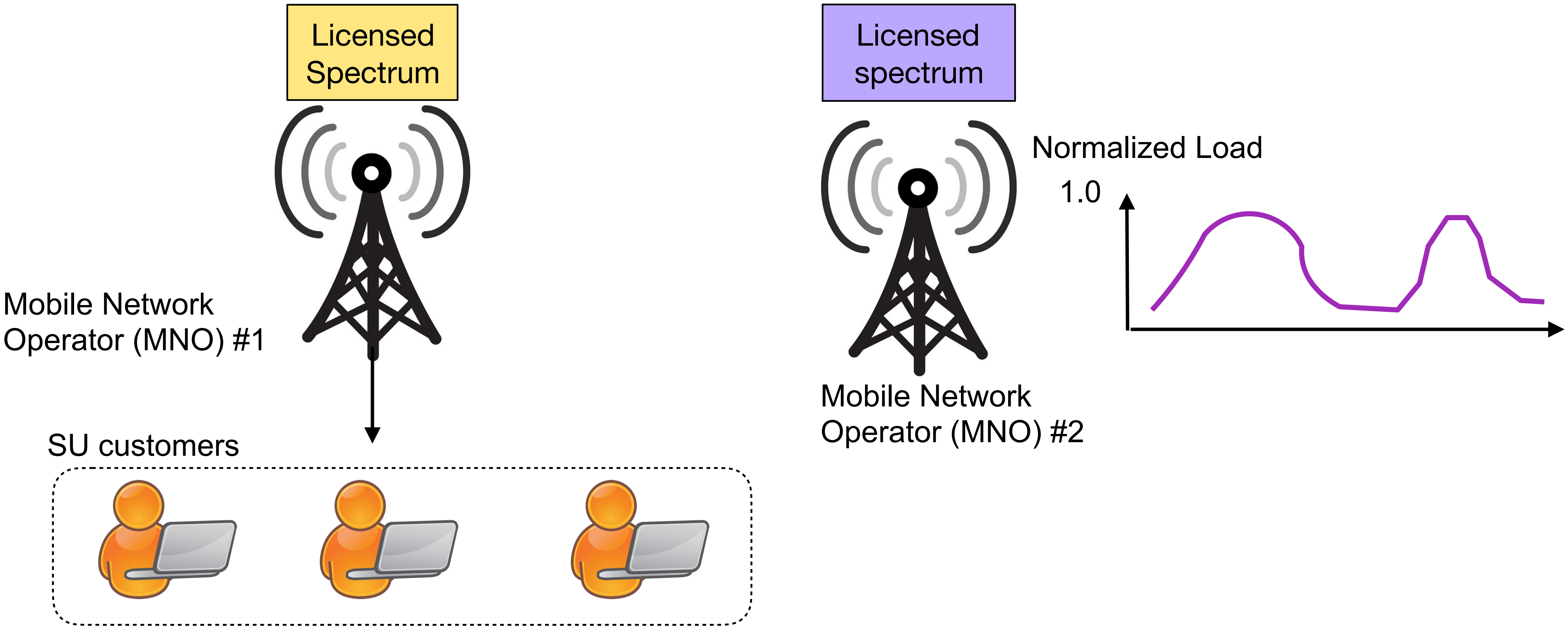
Source: https://tefficient.com/ Unlimited moves the needle – but it's when mobile addresses slow fixed internet that something happens

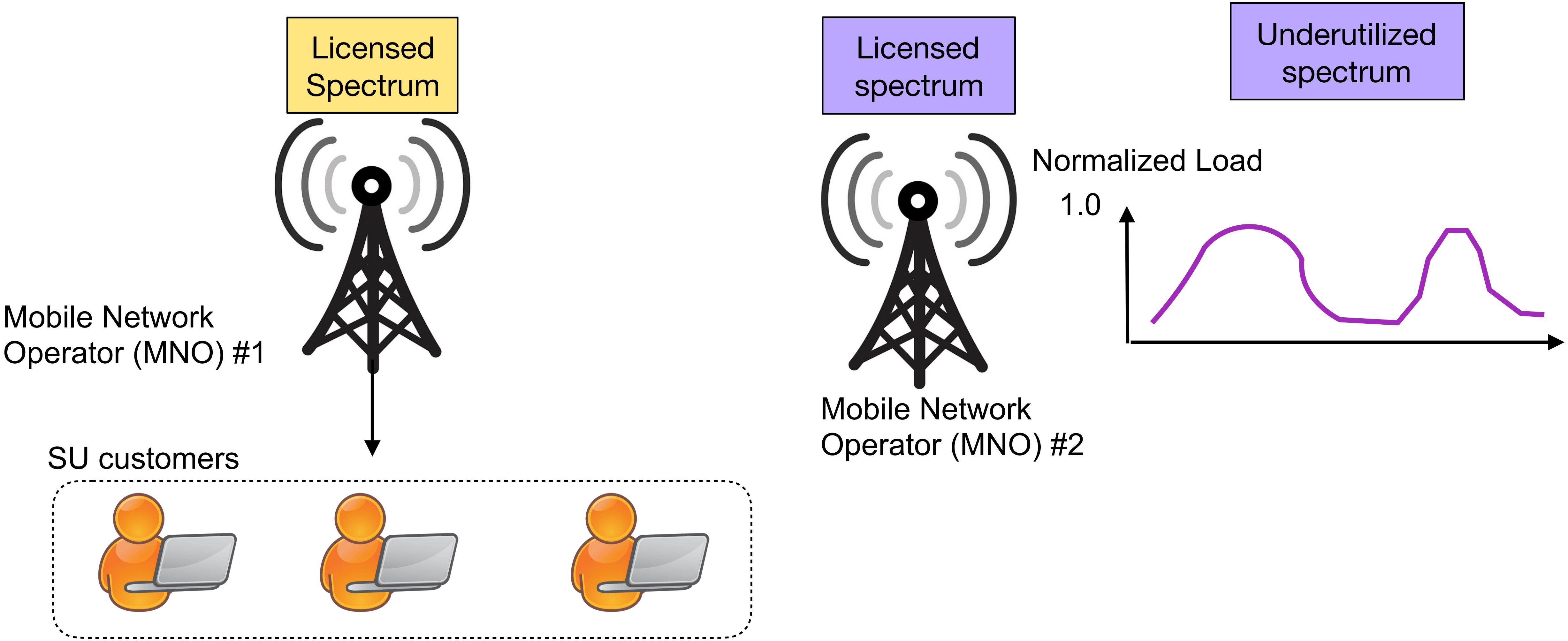


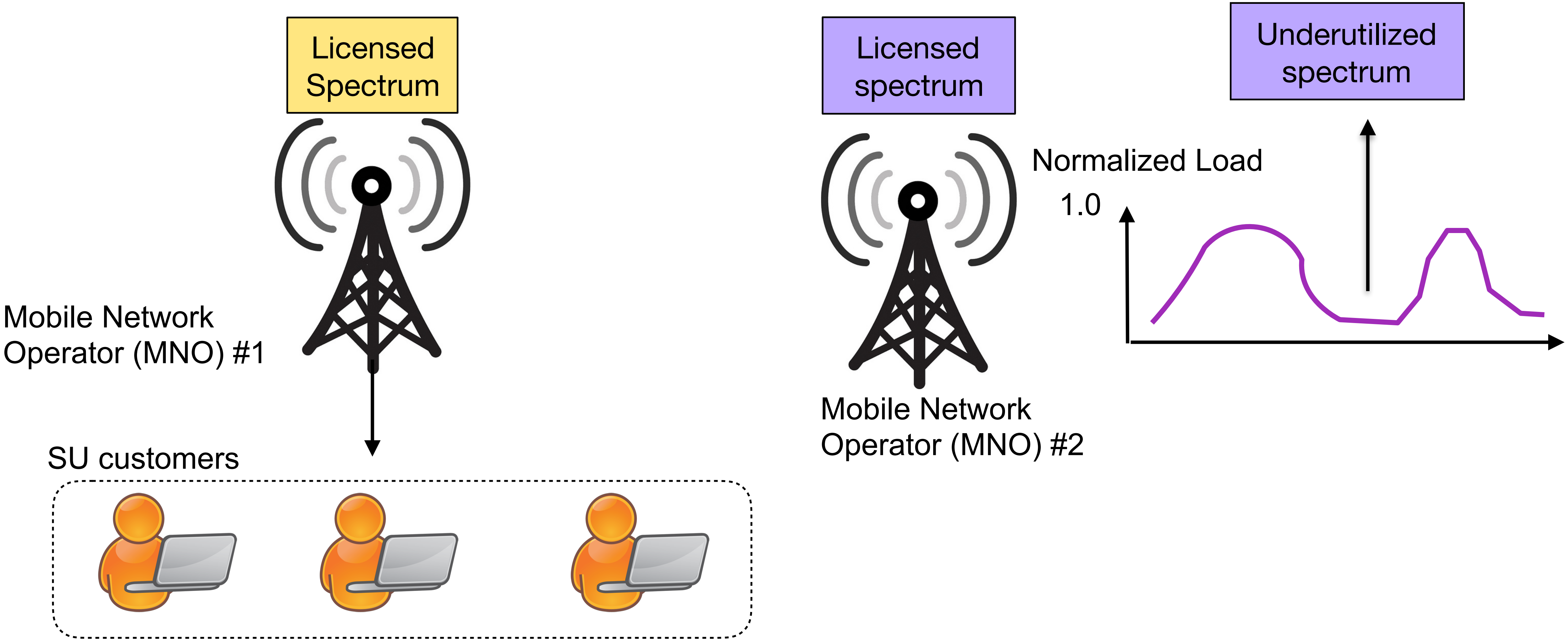

Licensed
Spectrum
Mobile Network
Operator (MNO) #1
SU customers

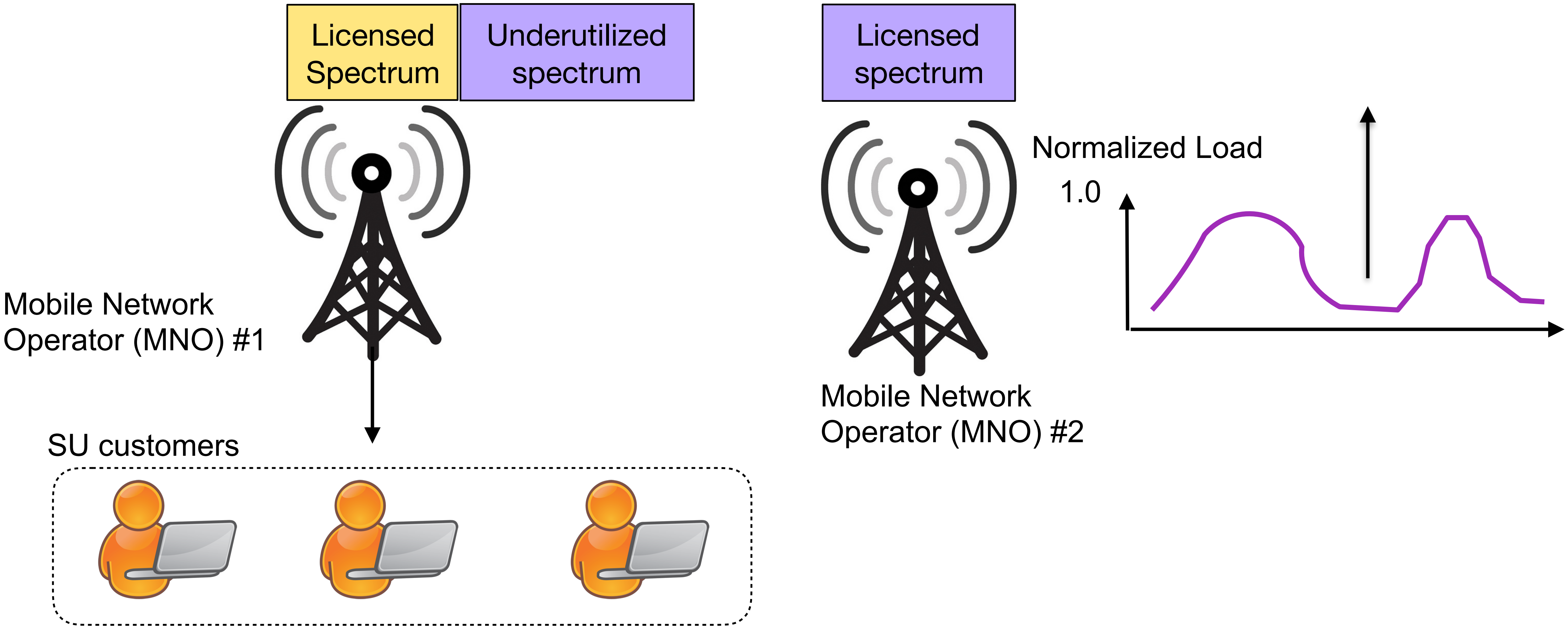

Licensed
Spectrum
Licensed
spectrum
Mobile Network
Operator (MNO) #1
Mobile Network
Operator (MNO) #2
SU customers

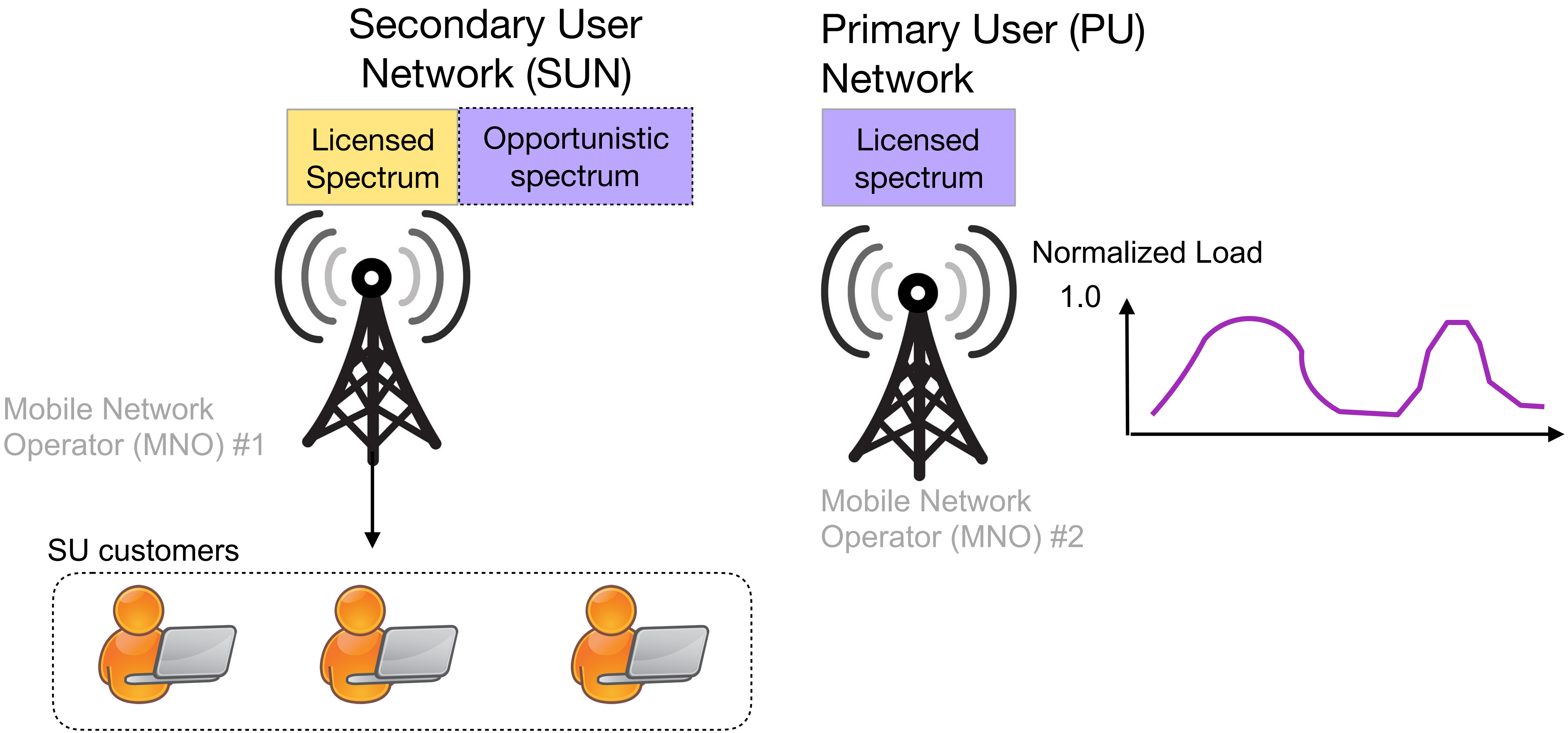

Licensed Spectrum
Licensed spectrum
Normalized Load
1.0
Mobile Network Operator (MNO) #1
Mobile Network Operator (MNO) #2
SU customers


Licensed Spectrum
Licensed spectrum
Normalized Load
1.0
Mobile Network Operator (MNO) #1
Mobile Network Operator (MNO) #2
SU customers

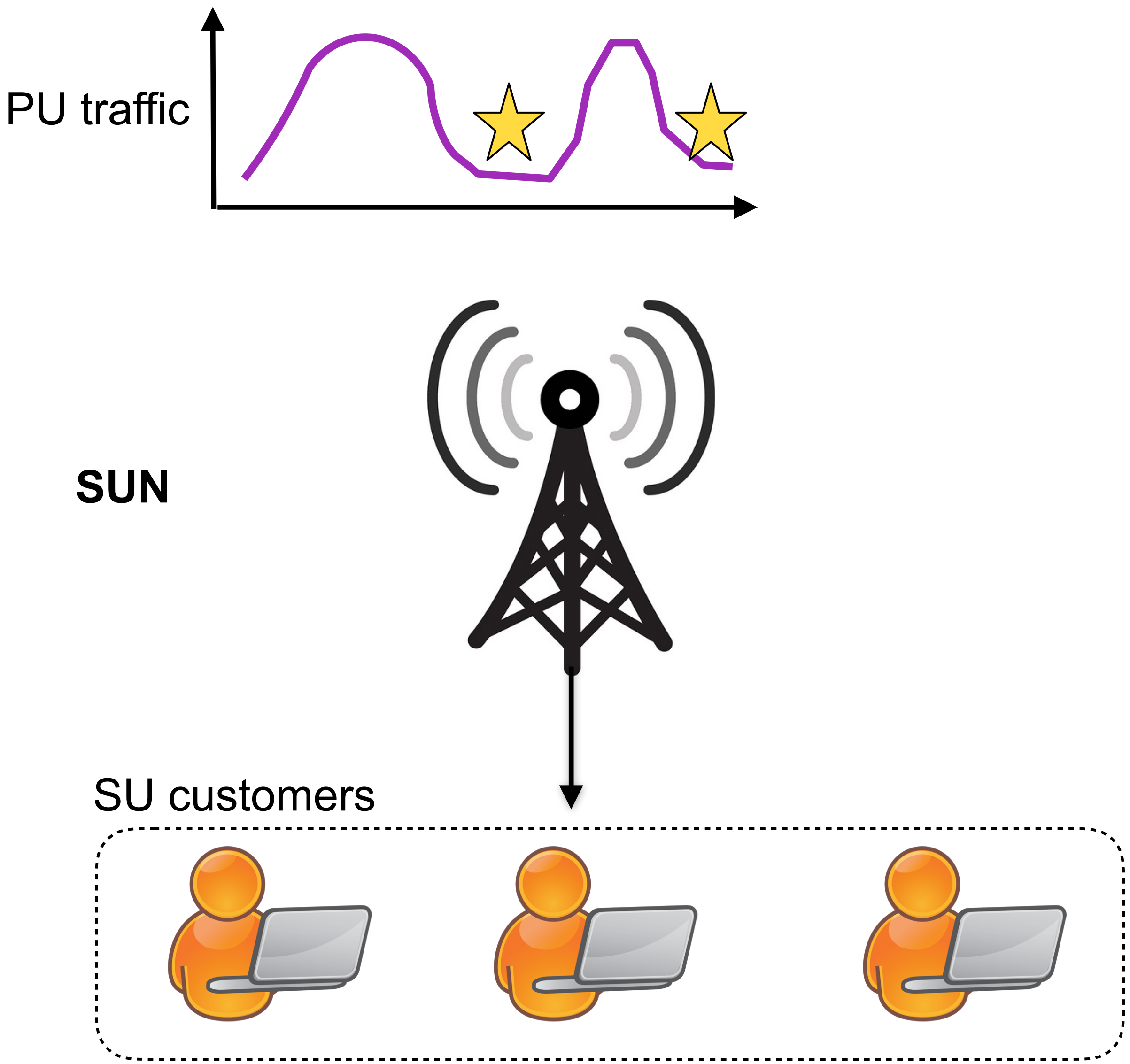

Licensed
Spectrum
Mobile Network
Operator (MNO) #1
SU customers
Licensed
spectrum
Underutilized
spectrum
Normalized Load
1.0
Mobile Network
Operator (MNO) #2


Licensed
Spectrum
Mobile Network
Operator (MNO) #1
SU customers
Licensed
spectrum
Underutilized
spectrum
Normalized Load
1.0
Mobile Network
Operator (MNO) #2

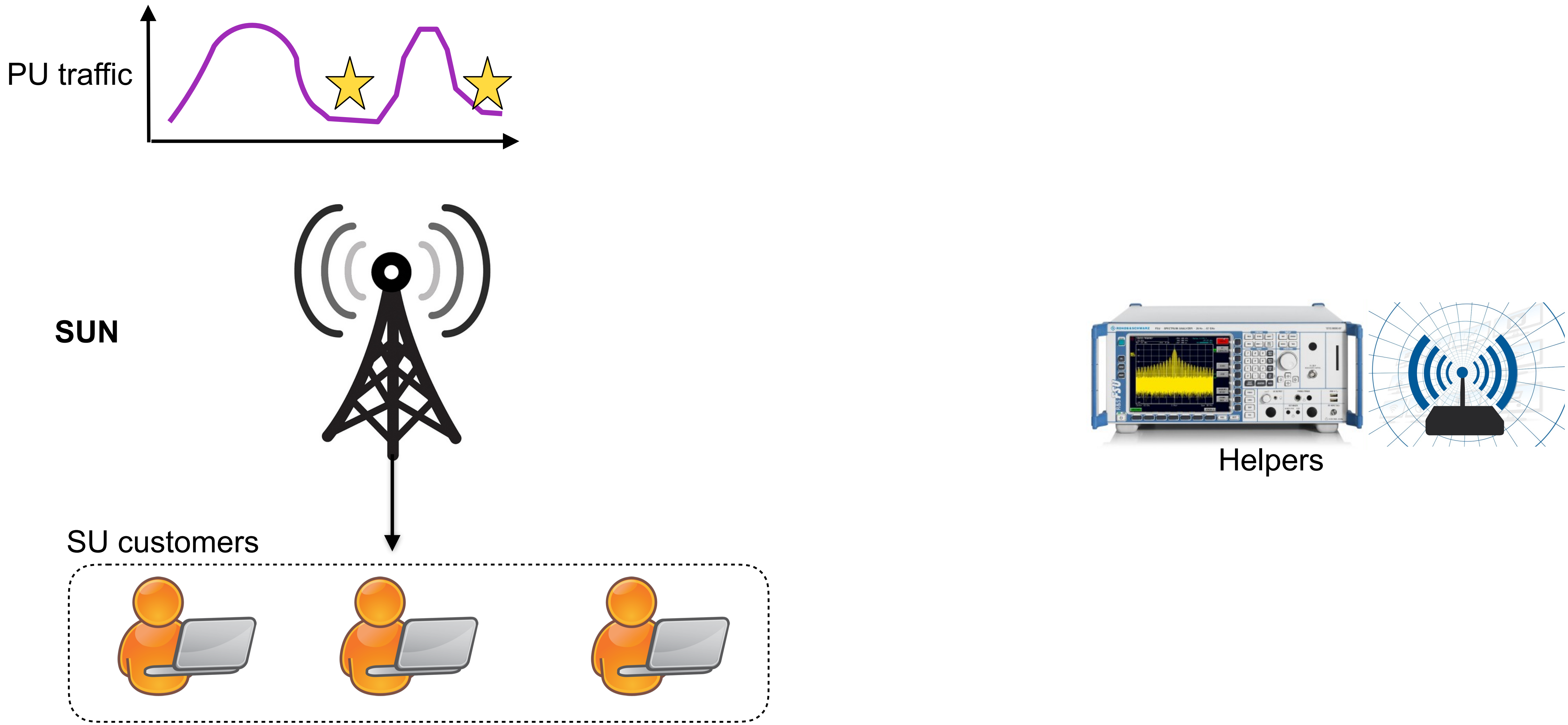

Licensed
Spectrum
Underutilized
spectrum
Licensed
spectrum
Normalized Load
1.0
Mobile Network
Operator (MNO) #1
Mobile Network
Operator (MNO) #2
SU customers


Secondary User Network (SUN)
Licensed Spectrum
Opportunistic spectrum
Mobile Network Operator (MNO) #1
SU customers
Primary User (PU) Network
Licensed spectrum
Normalized Load
1.0
Mobile Network Operator (MNO) #2

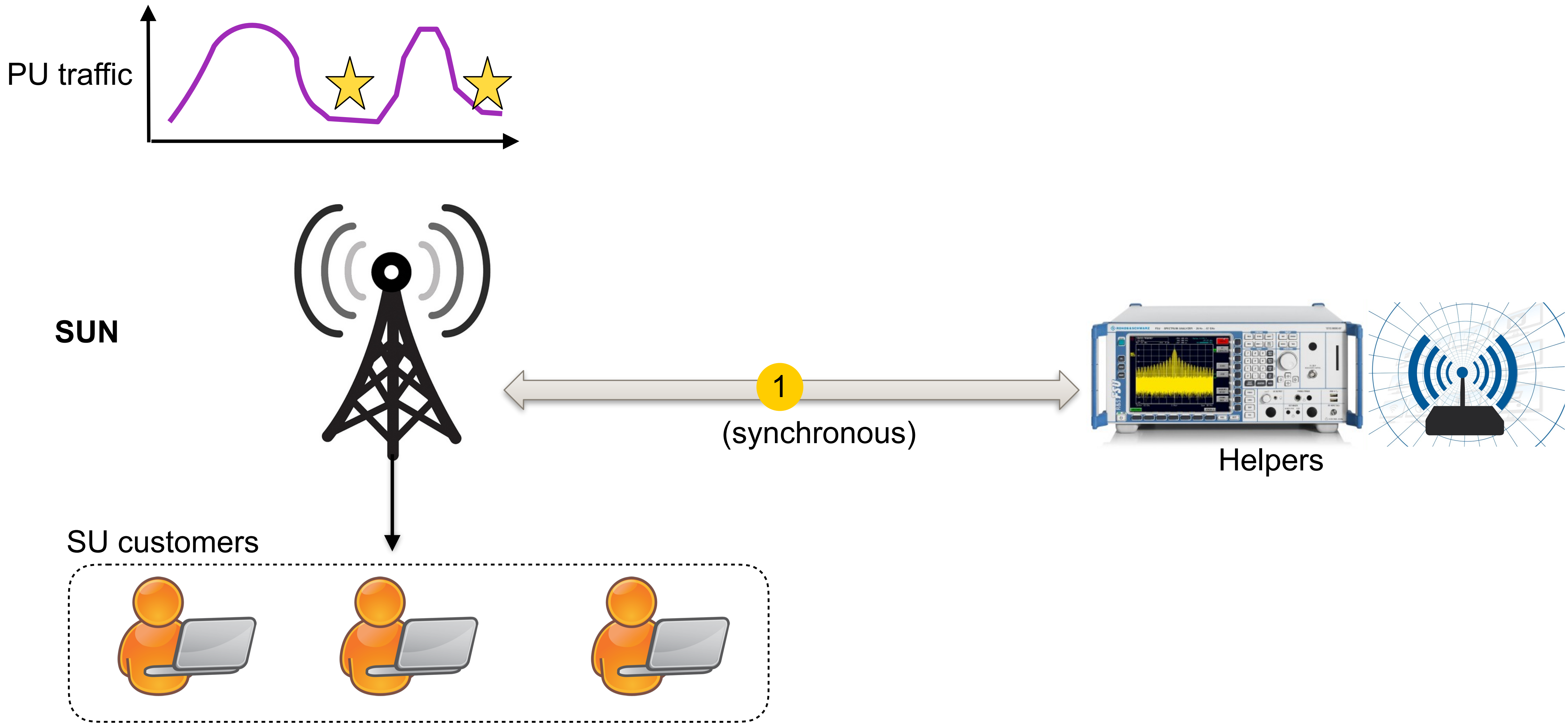
Crowdsourced spectrum discovery



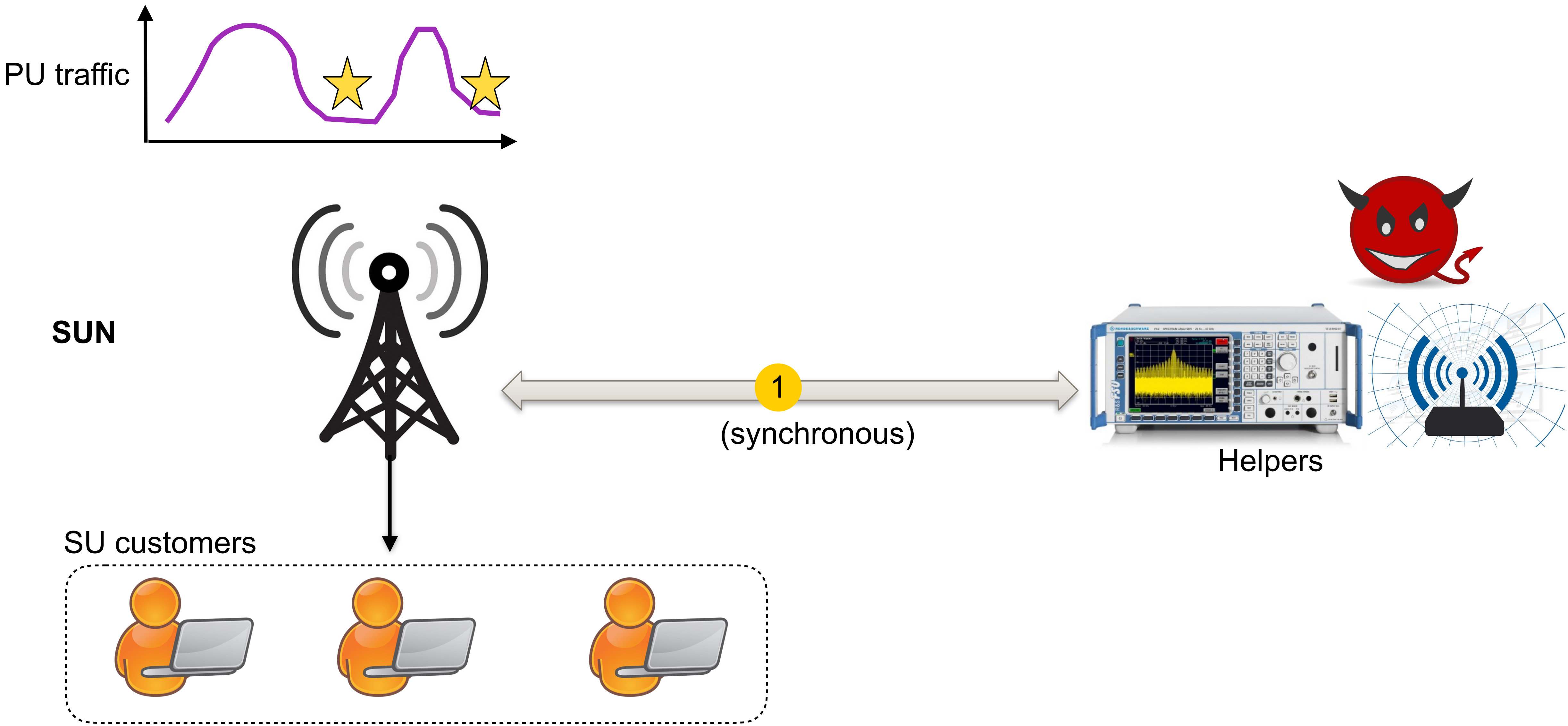
Crowdsourced spectrum discovery



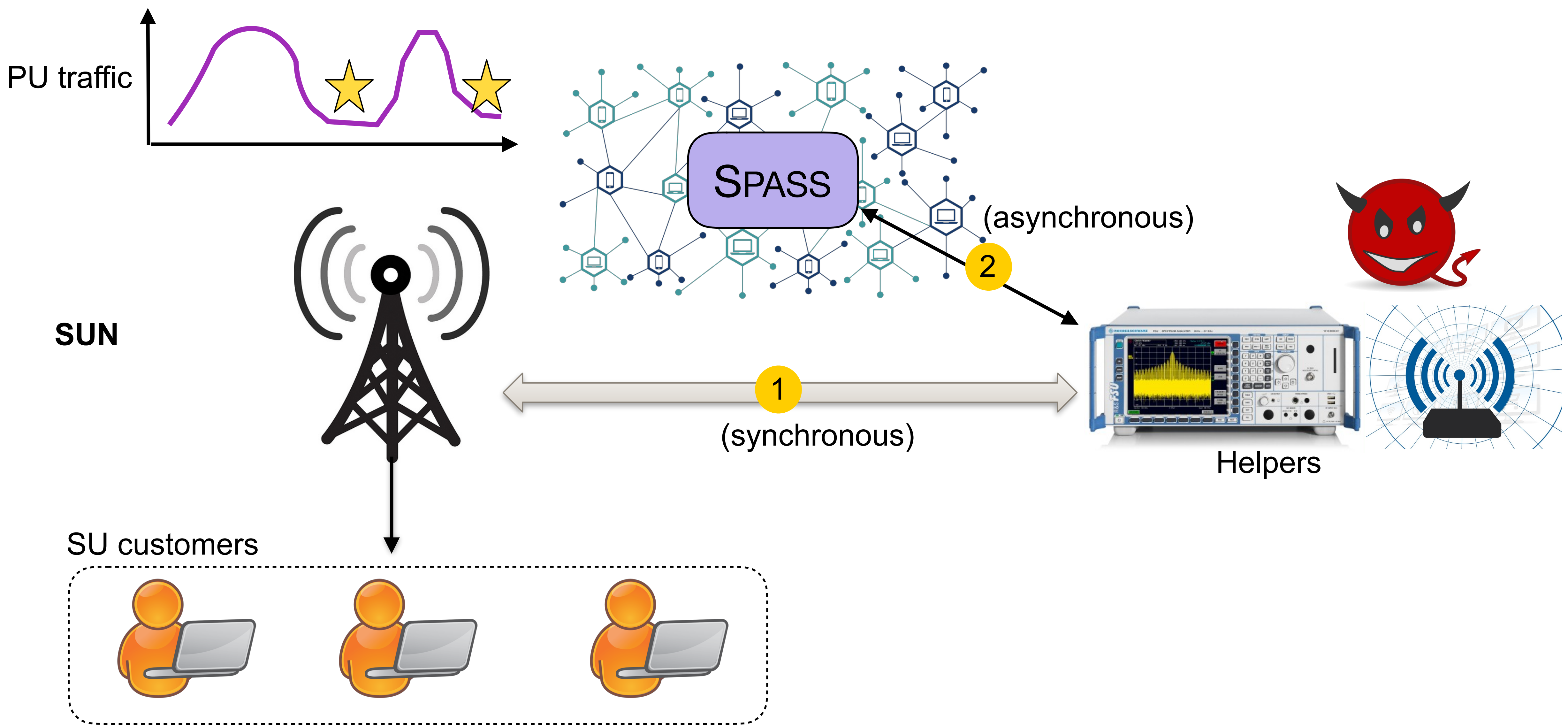
Crowdsourced spectrum discovery



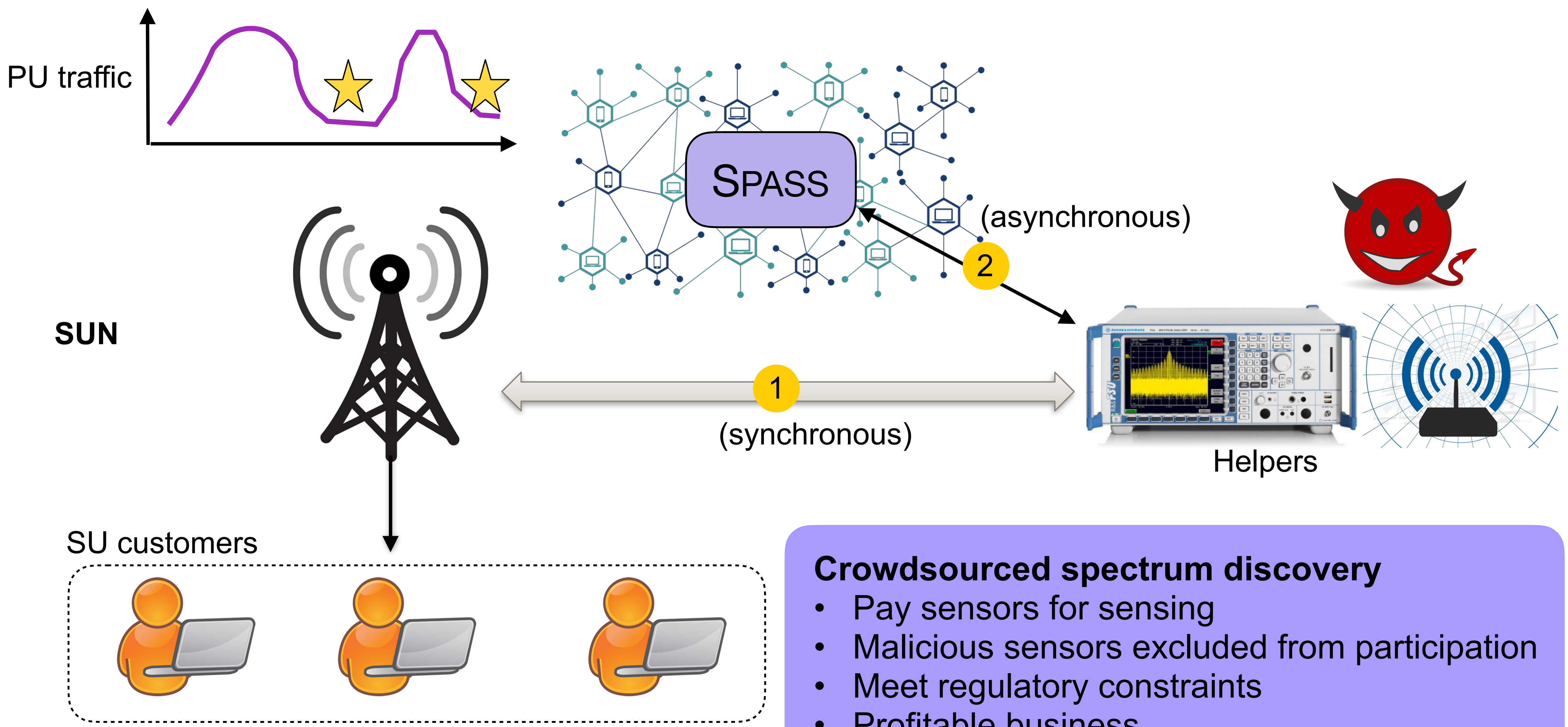
Crowdsourced spectrum discovery



Crowdsourced spectrum discovery

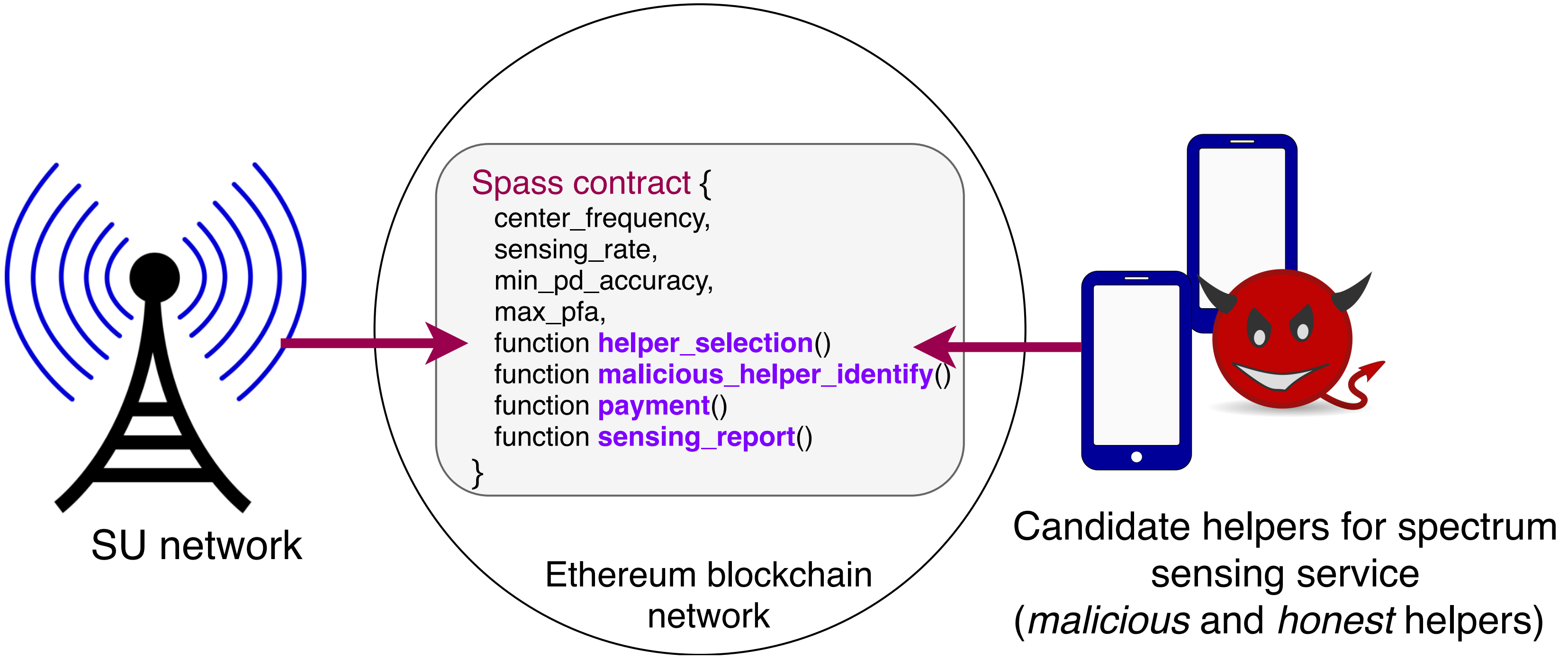


Crowdsourced spectrum discovery



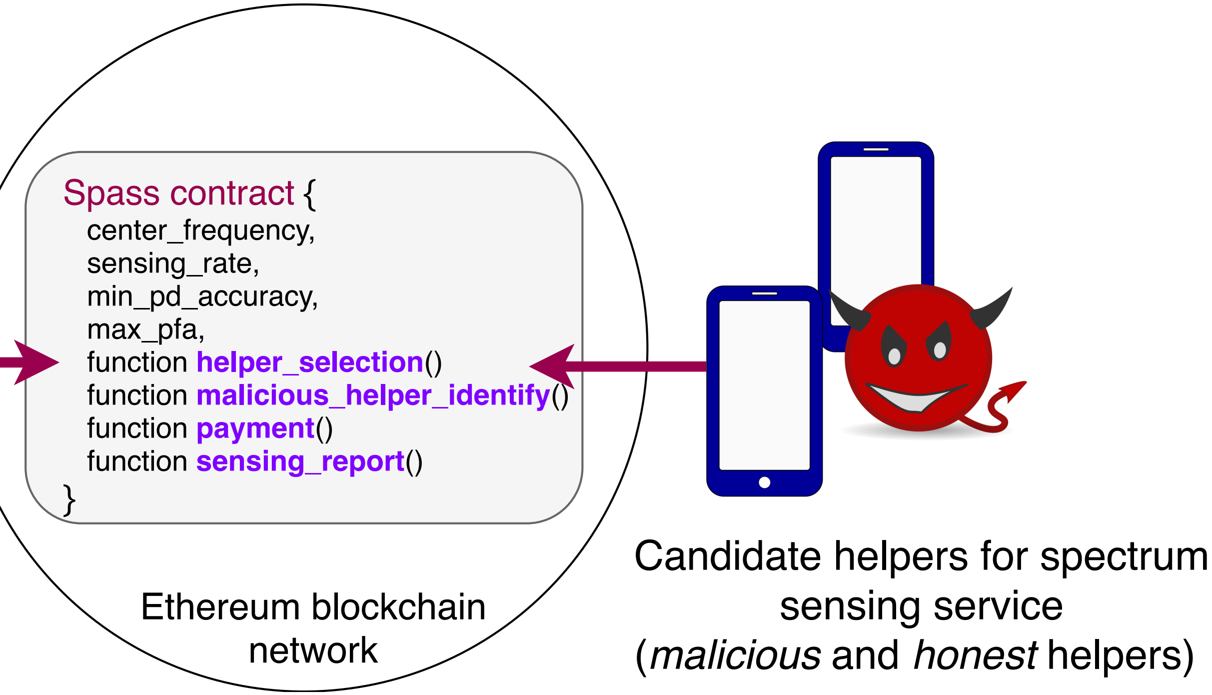
Crowdsourced spectrum discovery
- Pay sensors for sensing
- Malicious sensors excluded from participation
- Meet regulatory constraints
- Profitable business

A Spass contract



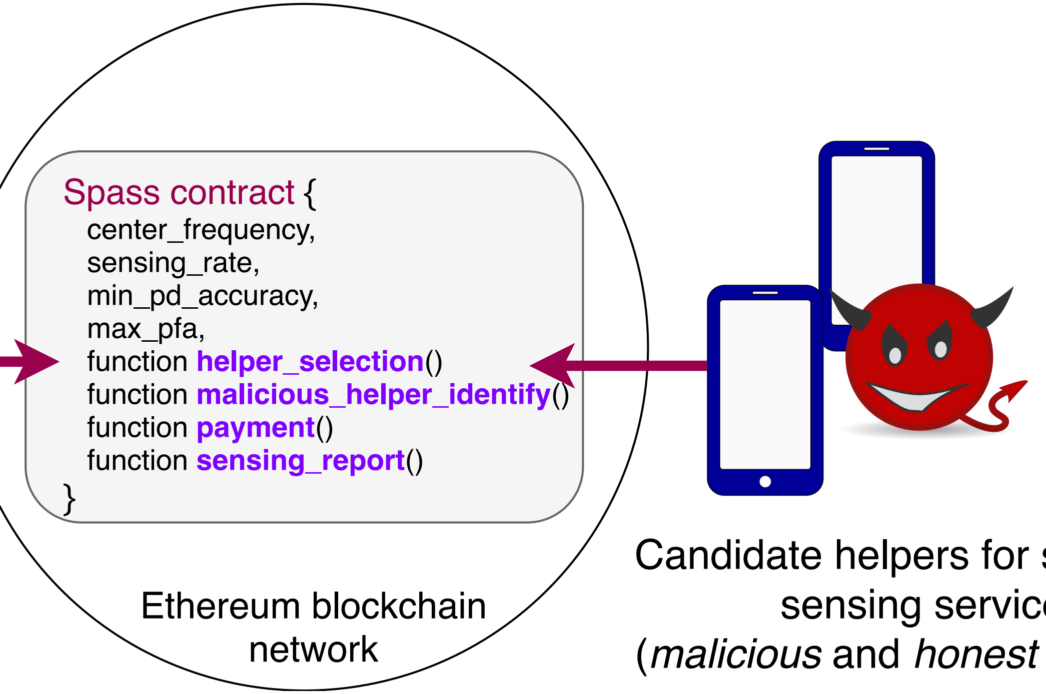
- Smart contract usage is not free
  - Write operations are expensive! (internal storage and manipulation of the contract)
  - Higher cost with increasing complexity of computation

A Spass contract



- Smart contract usage is not free
  - Write operations are expensive! (internal storage and manipulation of the contract)
  - Higher cost with increasing complexity of computation

A Spass contract



Candidate helpers for spectrum sensing service (*malicious* and *honest* helpers)

Operation	Gas	Description
ADD/SUB	3	Arithmetic operation
MUL/DIV	5	Arithmetic operation
ADDMOD/MULMOD	8	Arithmetic operation
AND/OR/XOR	3	Bitwise logic operation
LT/GT/SLT/SGT/EQ	3	Comparison operation
POP	2	Stack operation
PUSH/DUP/SWAP	3	Stack operation
MLOAD/MSTORE	3	Memory operation
JUMP	8	Unconditional jump
JUMPI	10	Conditional jump
SLOAD	200	Storage operation
SSTORE	5,000/20,000	Storage operation
BALANCE	400	Get balance of an account
CREATE	32,000	Create a new account using CREATE
CALL	25,000	Create a new account using CALL

Gas costs: https://docs.google.com/spreadsheets/d/1m89CVujrQe5LAFJ8-YAUCcNK950dUzMQPMJBxRtGCqs/edit#gid=0

- Smart contract usage is not free
  - Write operations are expensive! (internal storage and manipulation of the contract)
  - Higher cost with increasing complexity of computation

Our proposal:
Smart Contracts for Spectrum Sensing (Spass)

(i) what is the *cost of running smart contract* based spectrum discovery?

(ii) given that both the helpers and the miners have to be paid, under which conditions an MNO can sustain a *profitable business* via smart-contract based spectrum discovery?

(iii) How can the smart-contract catch *free-riders* among the sensing participants?

Spass

- Design goals
- Contract functionality
- Optimal contract parameters
- Malicious helper identification
- Performance
- Business feasibility of Spass

For a feasible business model

For a feasible business model

SU MNO

Regulators

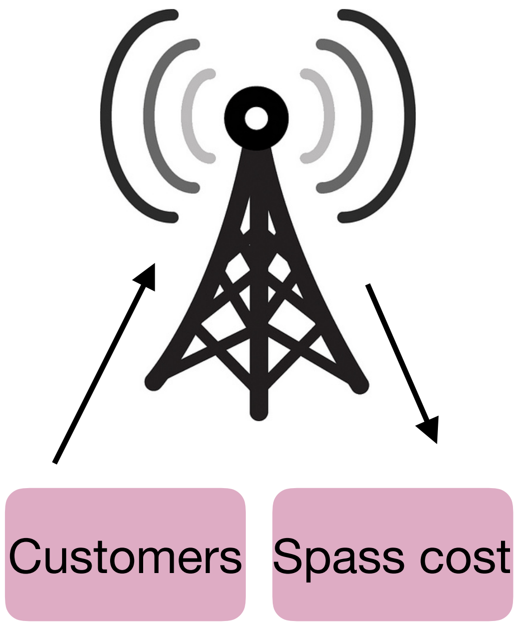
Helper nodes (sensors)

For a feasible business model

SU MNO

Regulators

Helper nodes (sensors)

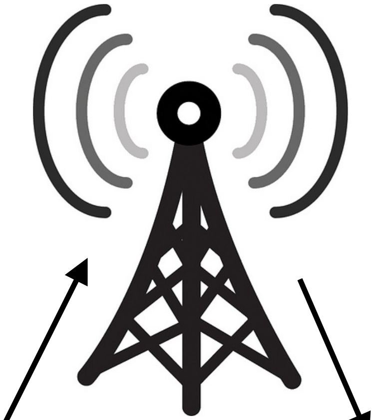


For a feasible business model

SU MNO

Regulators

Helper nodes (sensors)



Customers

Spass cost

Helpers
Sensing

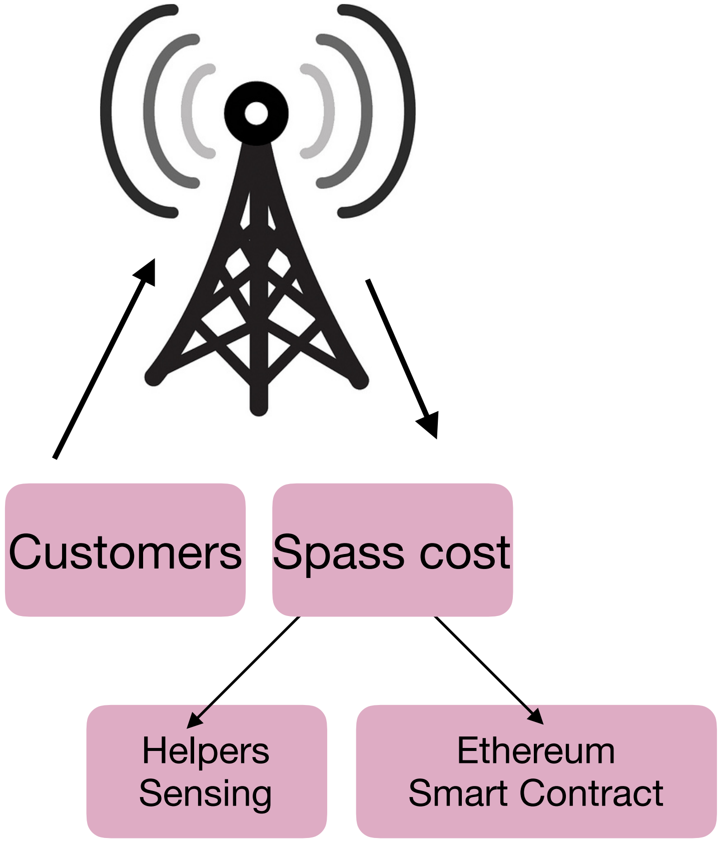
Ethereum
Smart Contract

For a feasible business model

SU MNO

Regulators

Helper nodes (sensors)



Viestintävirasto
Finnish Communications Regulatory Authority

Federal Communications Commission

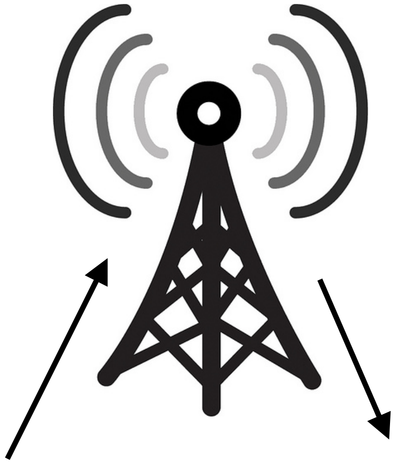
Ofcom

For a feasible business model

SU MNO

Regulators

Helper nodes (sensors)



Viestintävirasto
Finnish Communications
Regulatory Authority

Federal
Communications
Commission

Ofcom

Customers

Spass cost

Helpers
Sensing

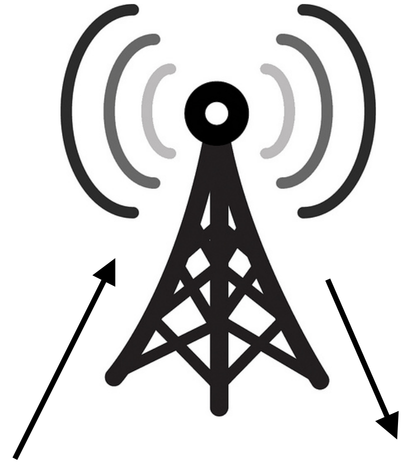
Ethereum
Smart Contract

High PU detection accuracy

Low false alarm in sensing

For a feasible business model

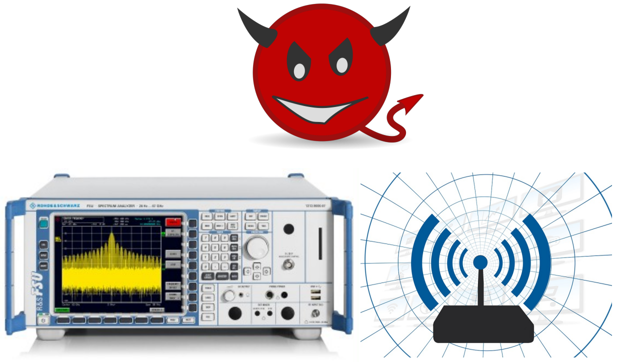
SU MNO	Regulators	Helper nodes (sensors)



Viestintävirasto
Finnish Communications
Regulatory Authority

Federal
Communications
Commission

Ofcom



Customers

Spass cost

Helpers Sensing

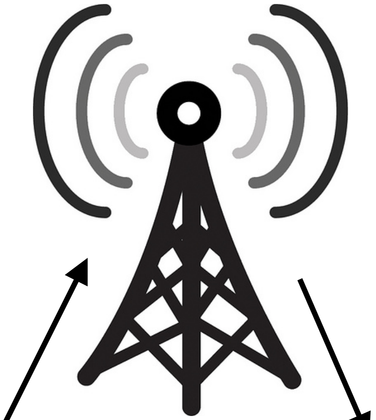
Ethereum Smart Contract

High PU detection accuracy

Low false alarm in sensing

For a feasible business model

SU MNO



Customers

Spass cost

Helpers Sensing

Ethereum Smart Contract

Regulators

Viestintävirasto
Finnish Communications Regulatory Authority

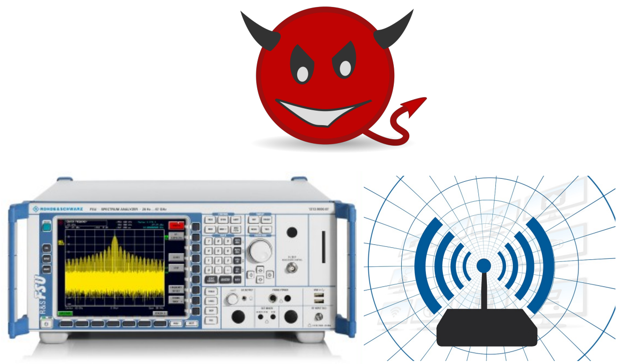
FCC Federal Communications Commission

Ofcom

High PU detection accuracy

Low false alarm in sensing

Helper nodes (sensors)

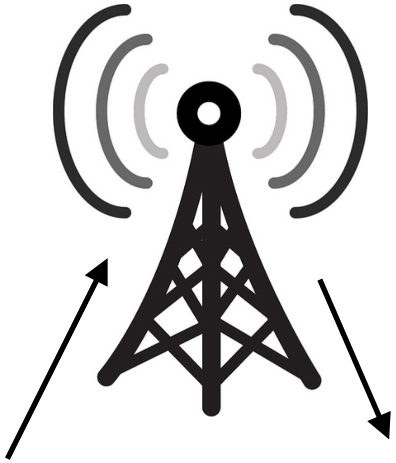


Attracting honest helpers

Able to catch malicious helpers (free riders)

For a feasible business model

SU MNO



Customers

Spass cost

Helpers Sensing

Ethereum Smart Contract

Regulators

Viestintävirasto
Finnish Communications
Regulatory Authority

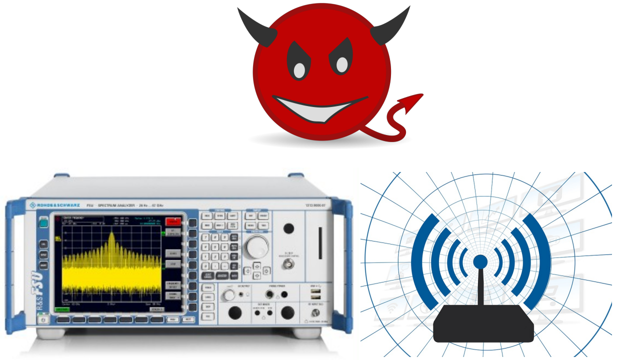
FCC Federal Communications Commission

Ofcom

High PU detection accuracy

Low false alarm in sensing

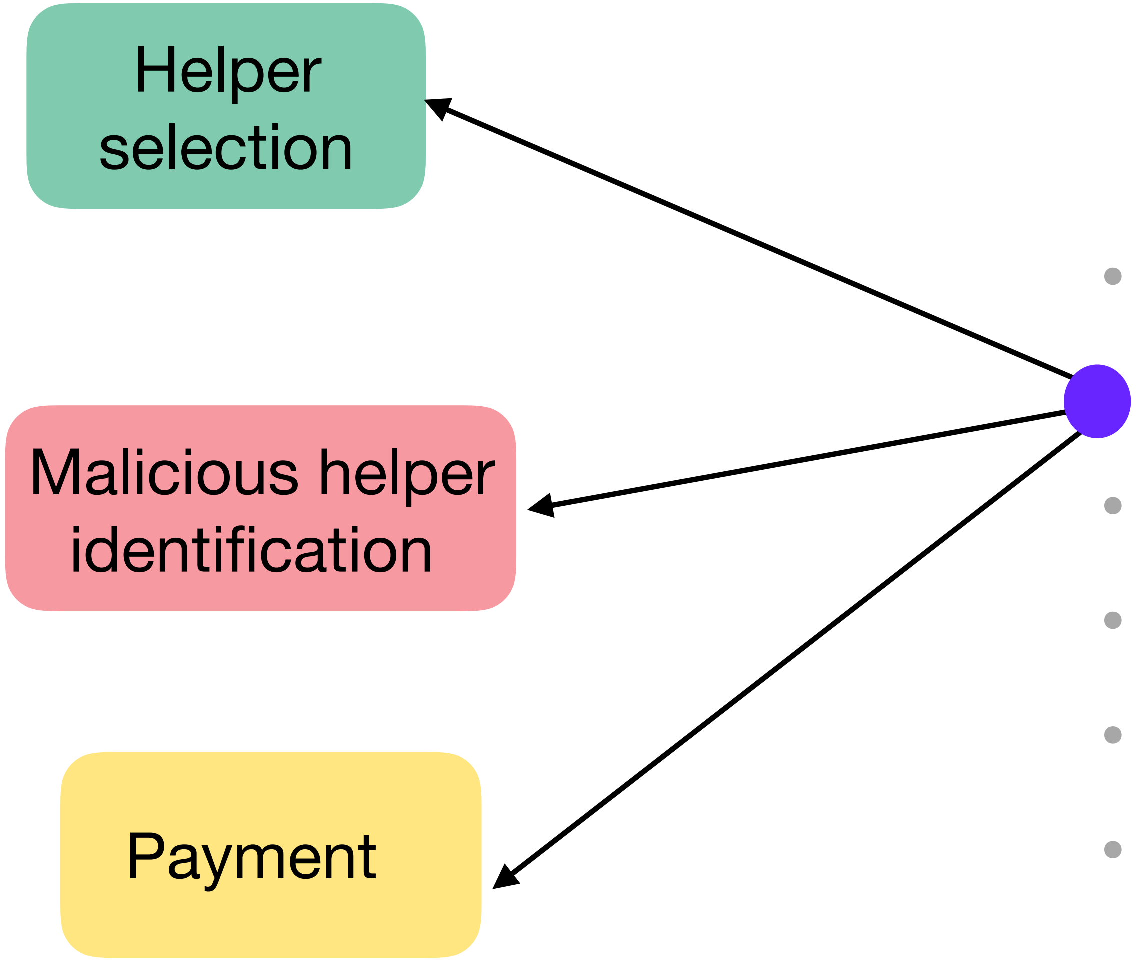
Helper nodes (sensors)



Attracting honest helpers

Able to catch malicious helpers (free riders)

This talk

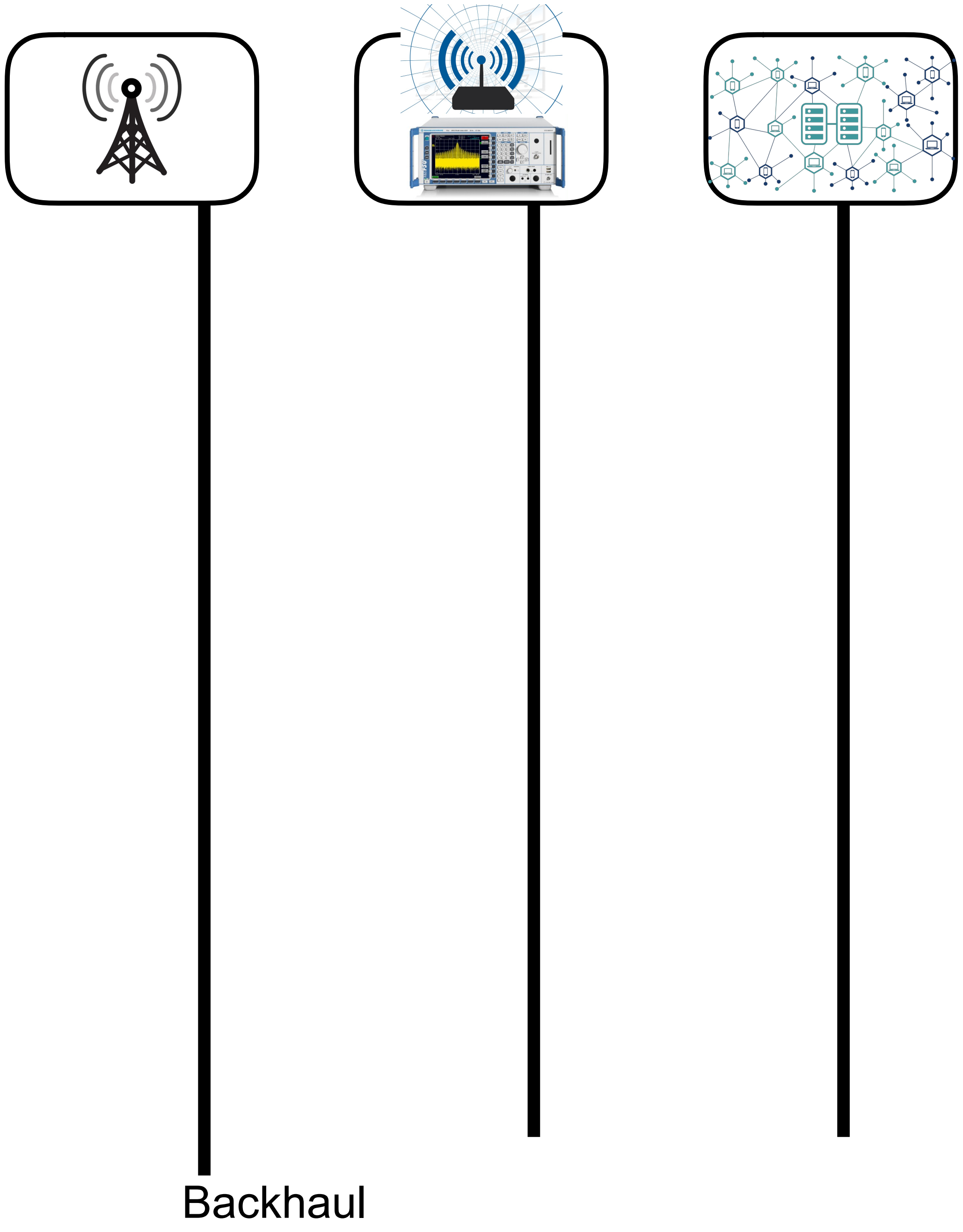


Spass

- Design goals
- **Contract functionality**
- Optimal contract parameters
- Malicious helper identification
- Performance
- Business feasibility of Spass

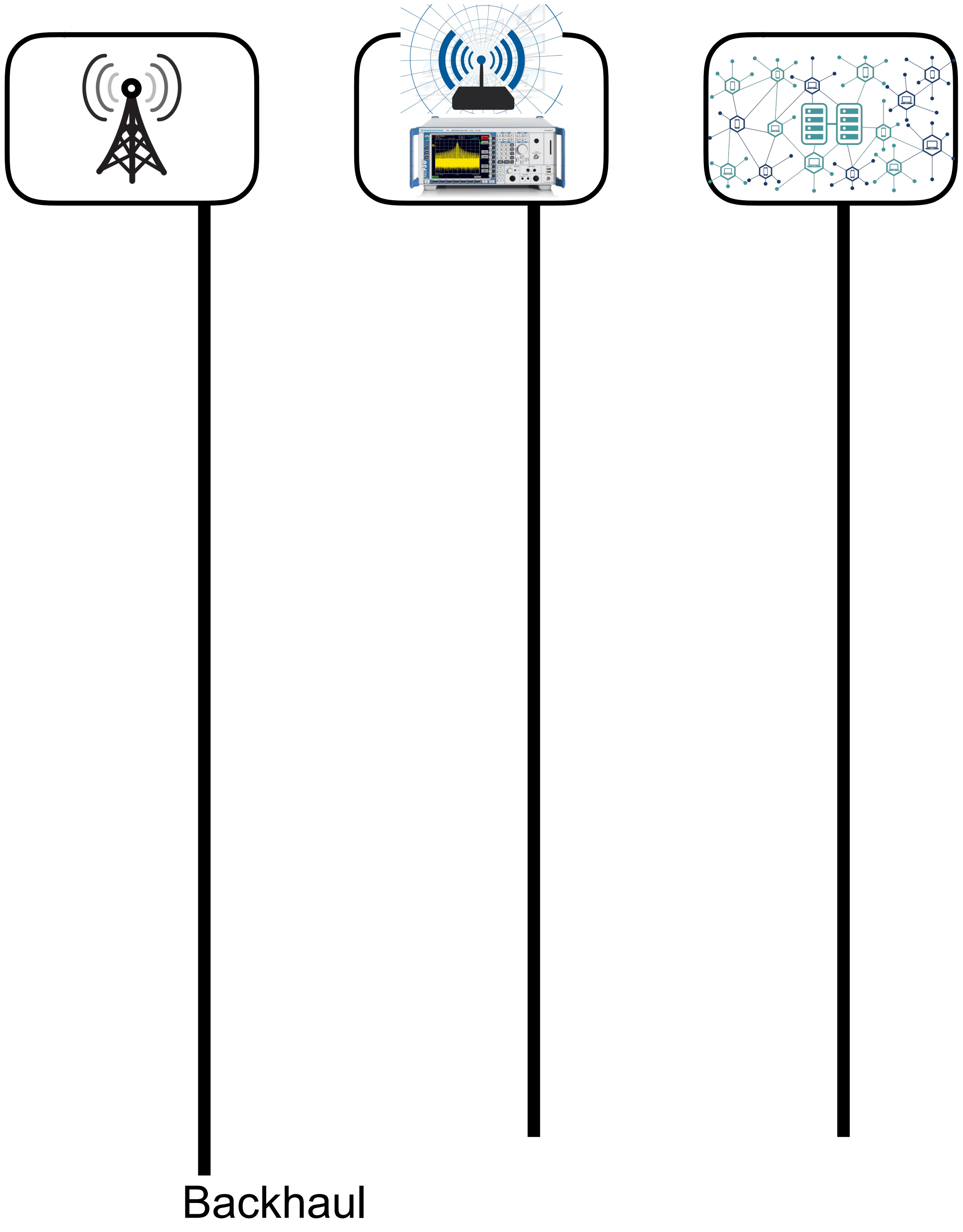
Message flow

- Contract duration: a certain time period



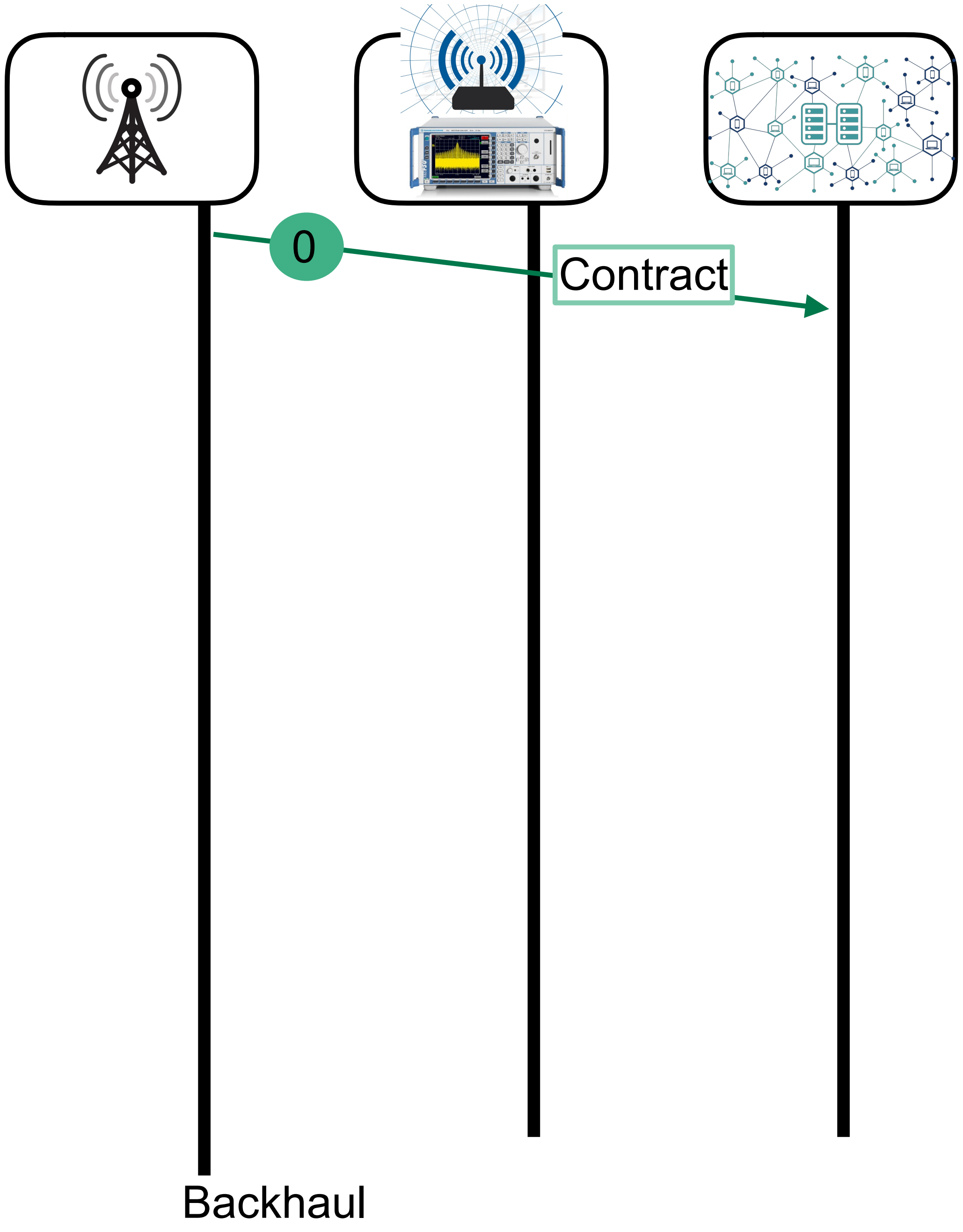
Message flow

- Contract duration: a certain time period
- SUN uploads the contract to Ethereum and receives a unique ETH address
- SUN broadcasts the contract address over the air



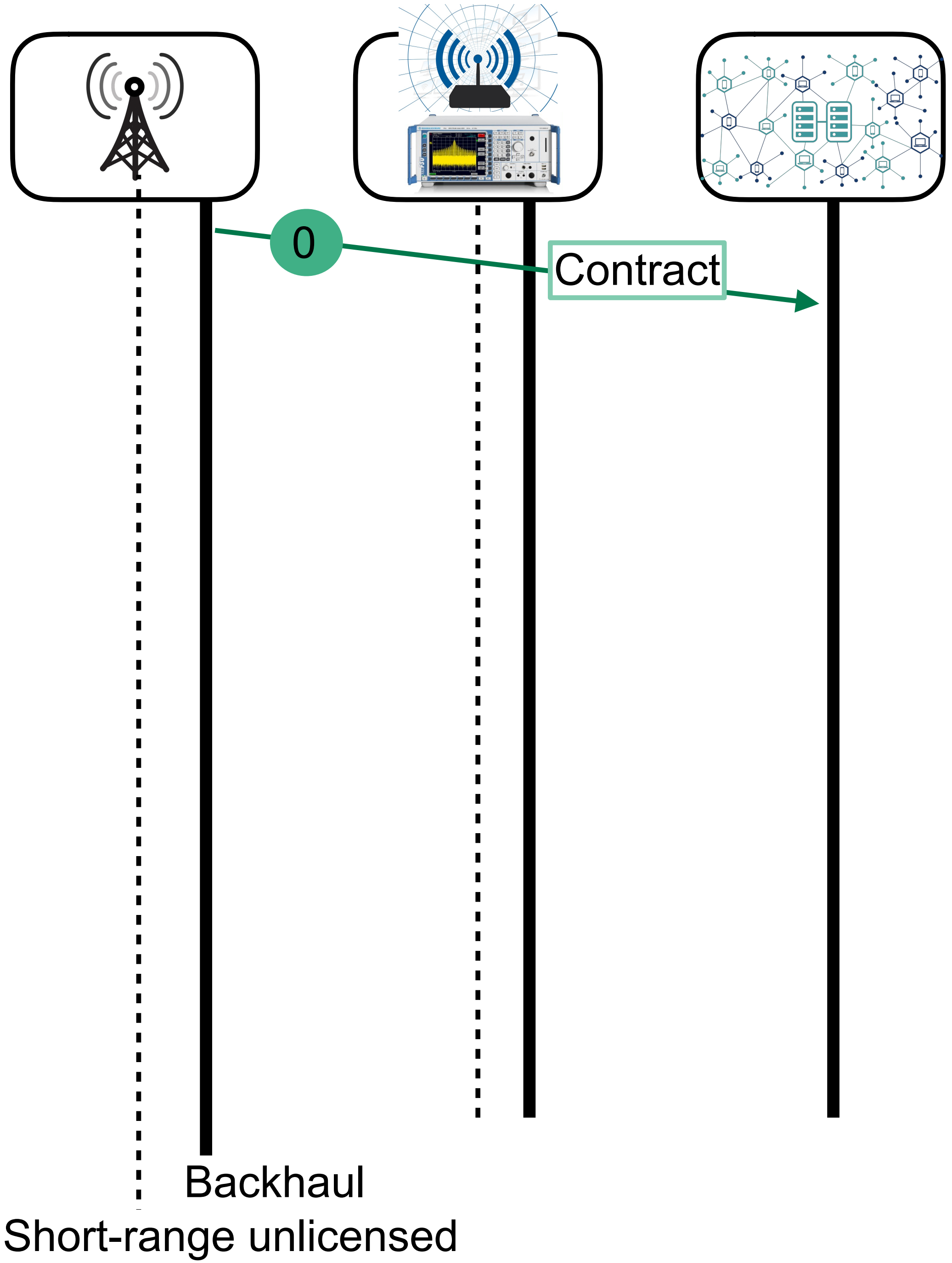
Message flow

- Contract duration: a certain time period
- SUN uploads the contract to Ethereum and receives a unique ETH address
- SUN broadcasts the contract address over the air



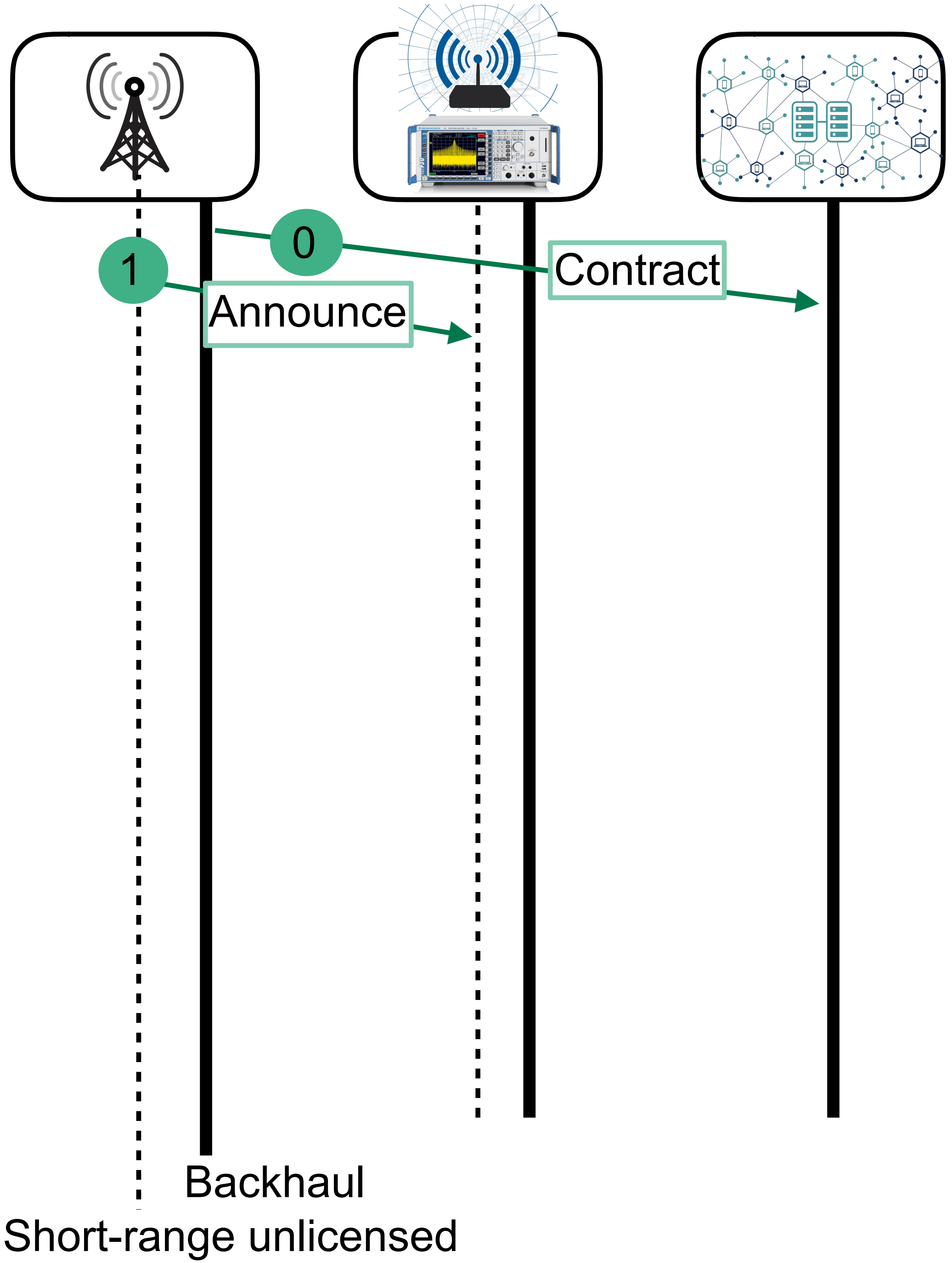
Message flow

- Contract duration: a certain time period
- SUN uploads the contract to Ethereum and receives a unique ETH address
- SUN broadcasts the contract address over the air

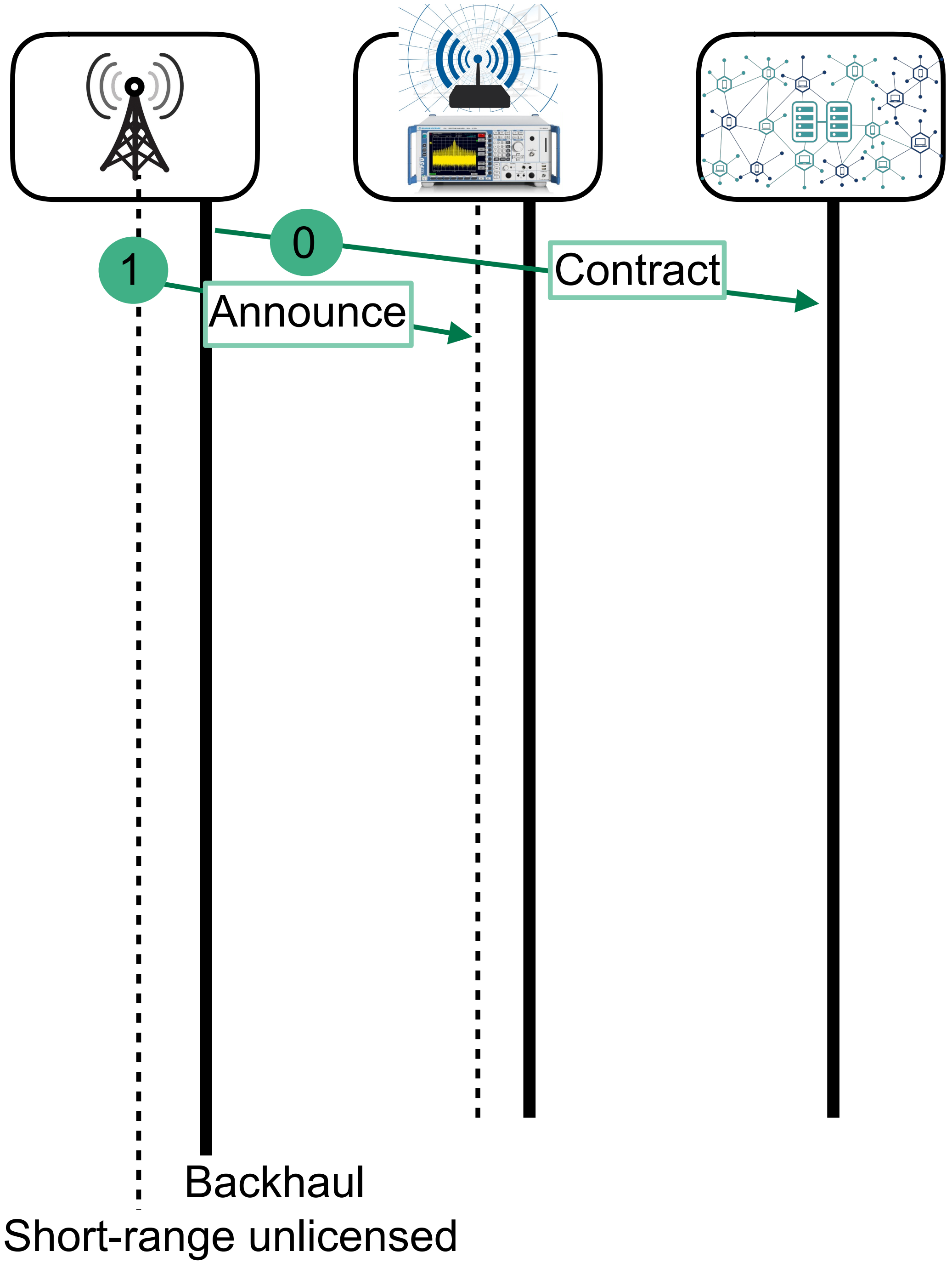


Message flow

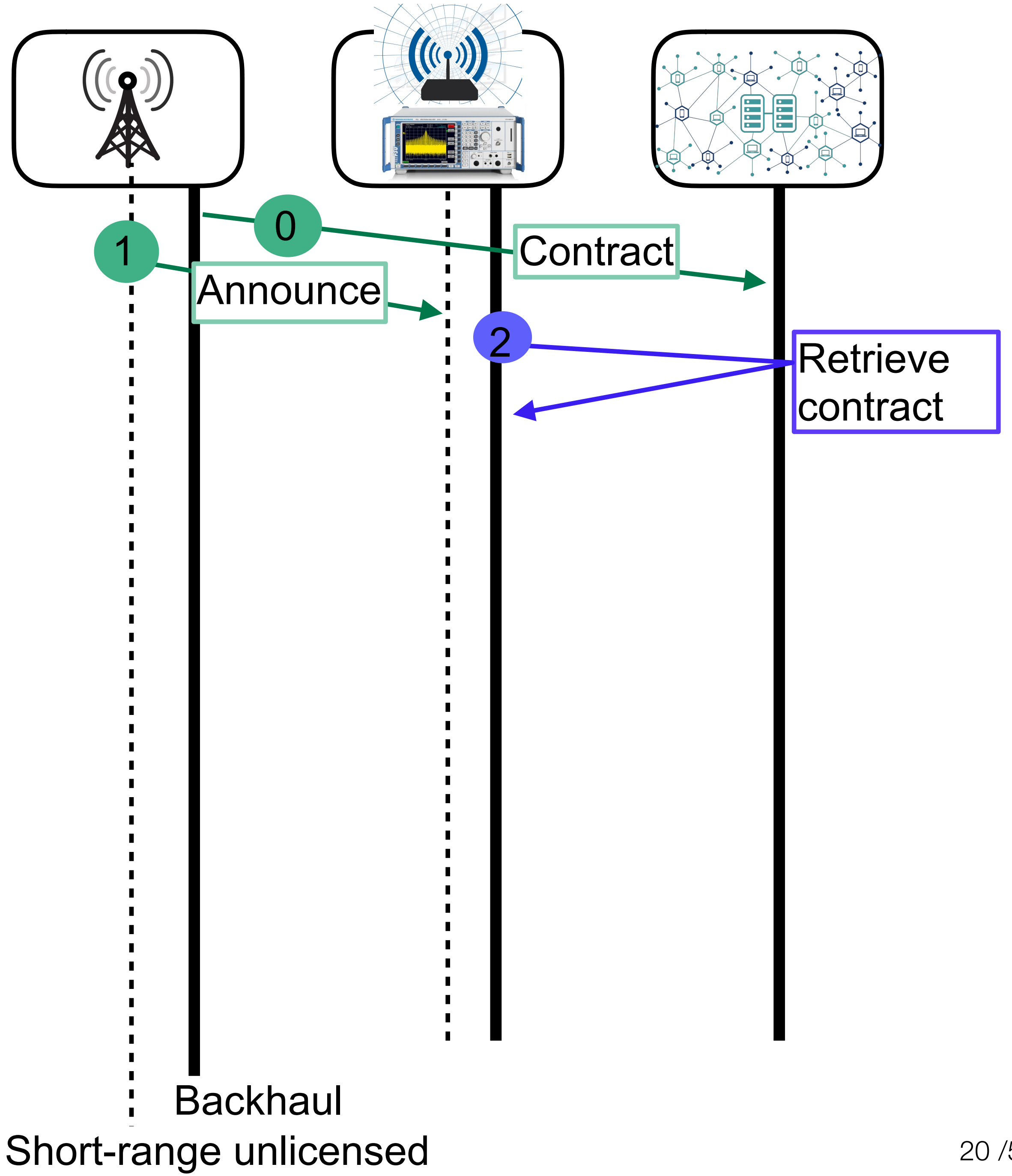
- Contract duration: a certain time period
- SUN uploads the contract to Ethereum and receives a unique ETH address
- SUN broadcasts the contract address over the air



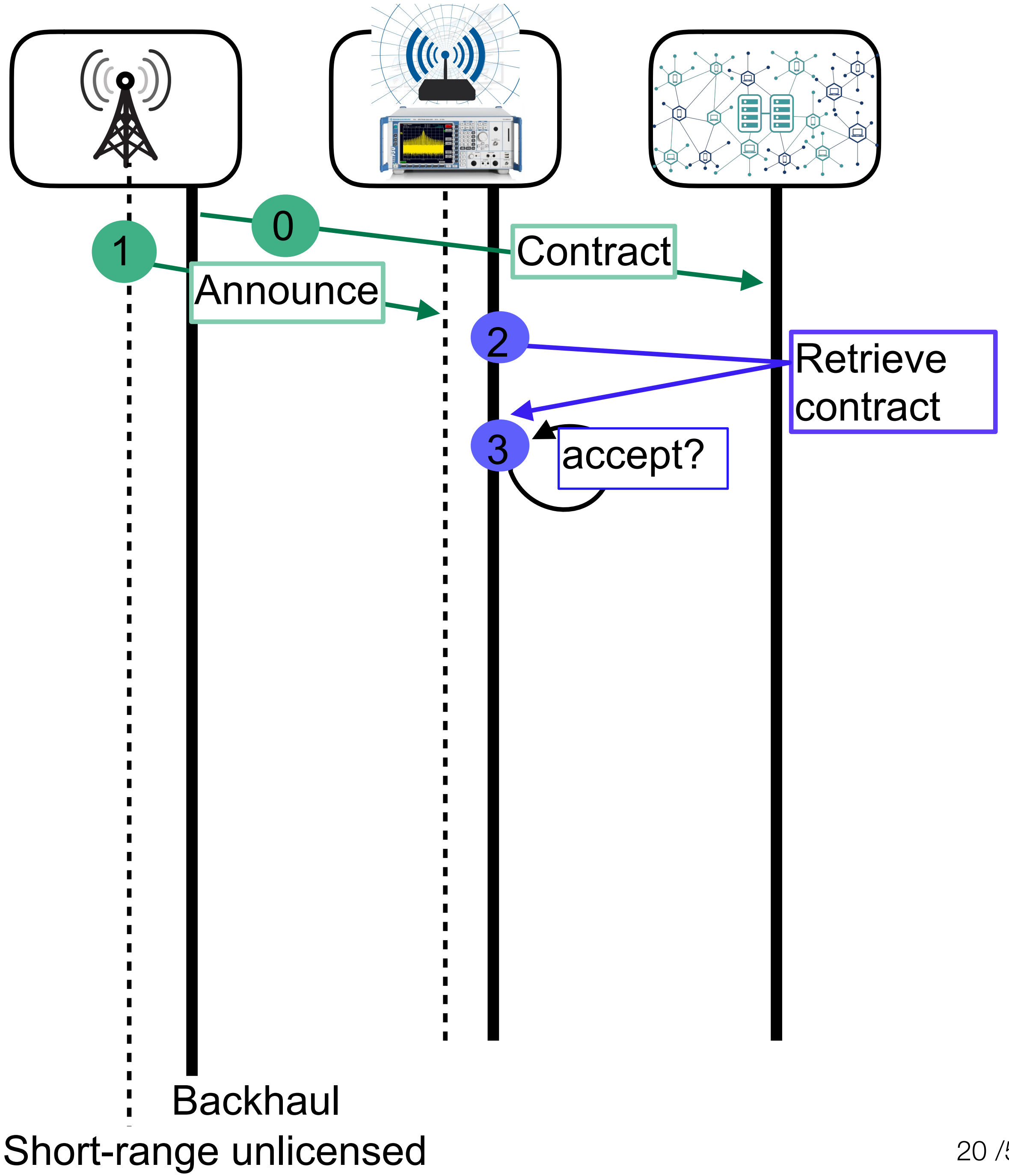
Message flow



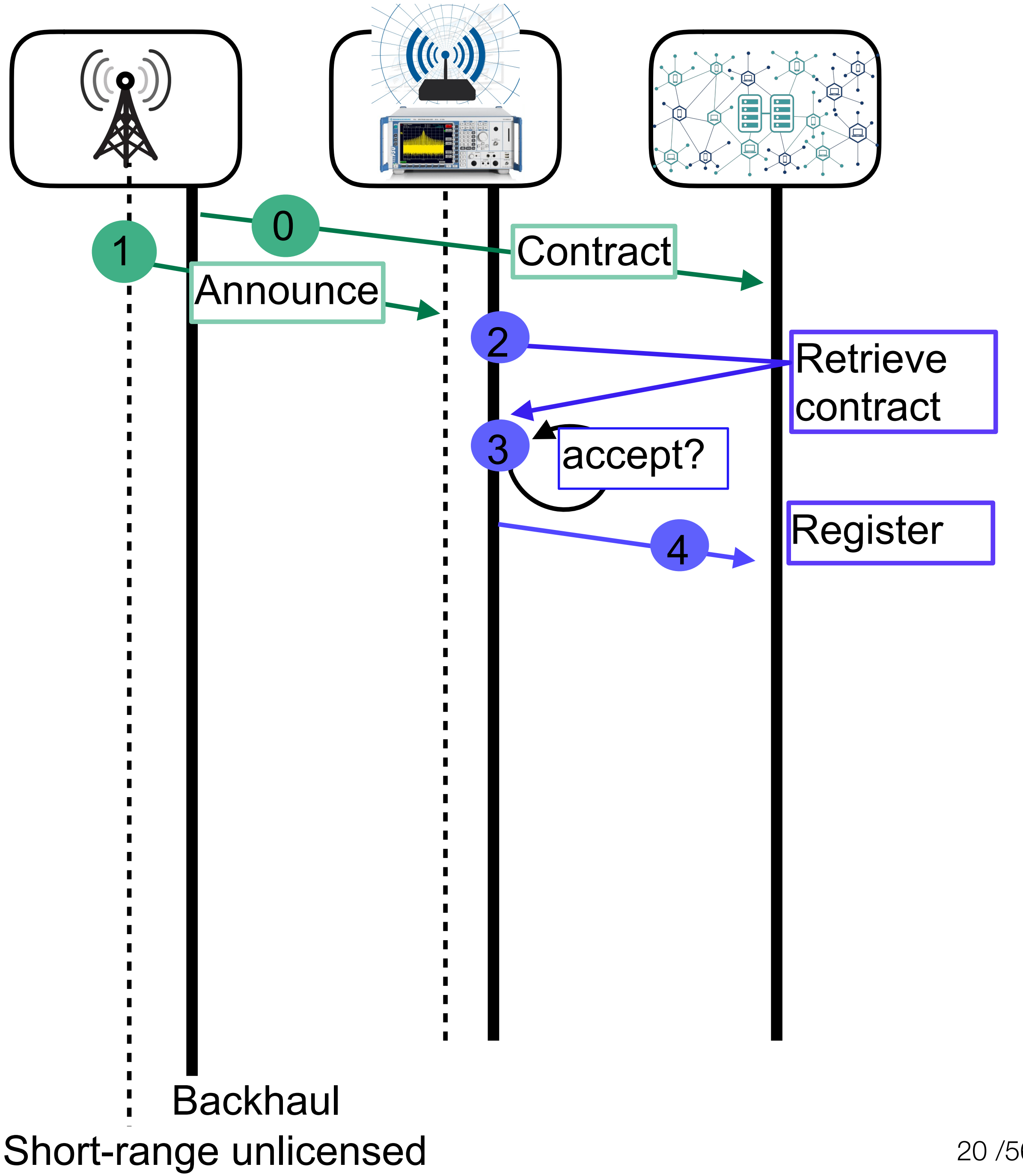
Message flow



Message flow

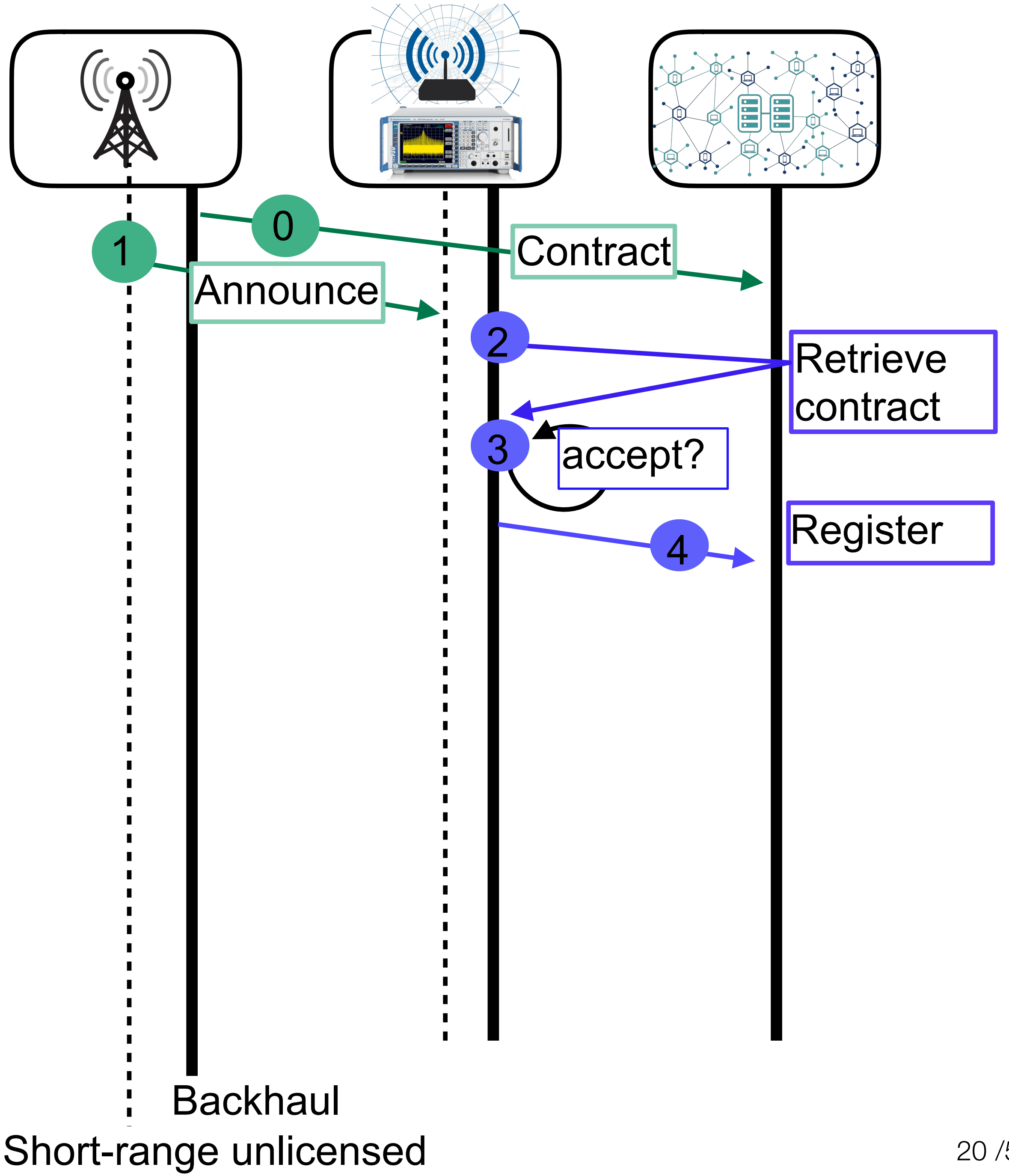


Message flow



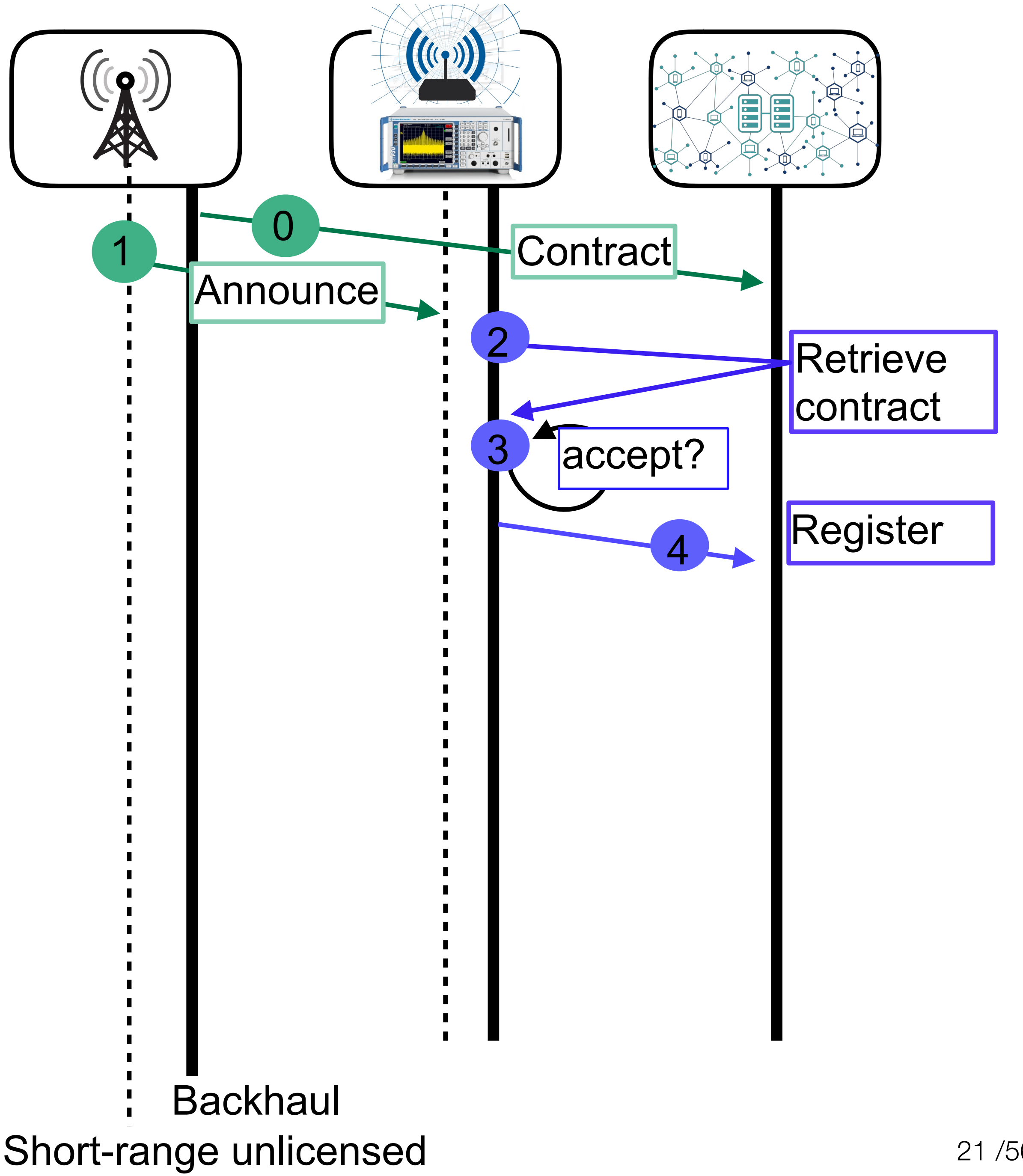
Message flow

- Helpers register:
    - accuracy in PU detection, false alarms
    - Sensing service price, location, etc.



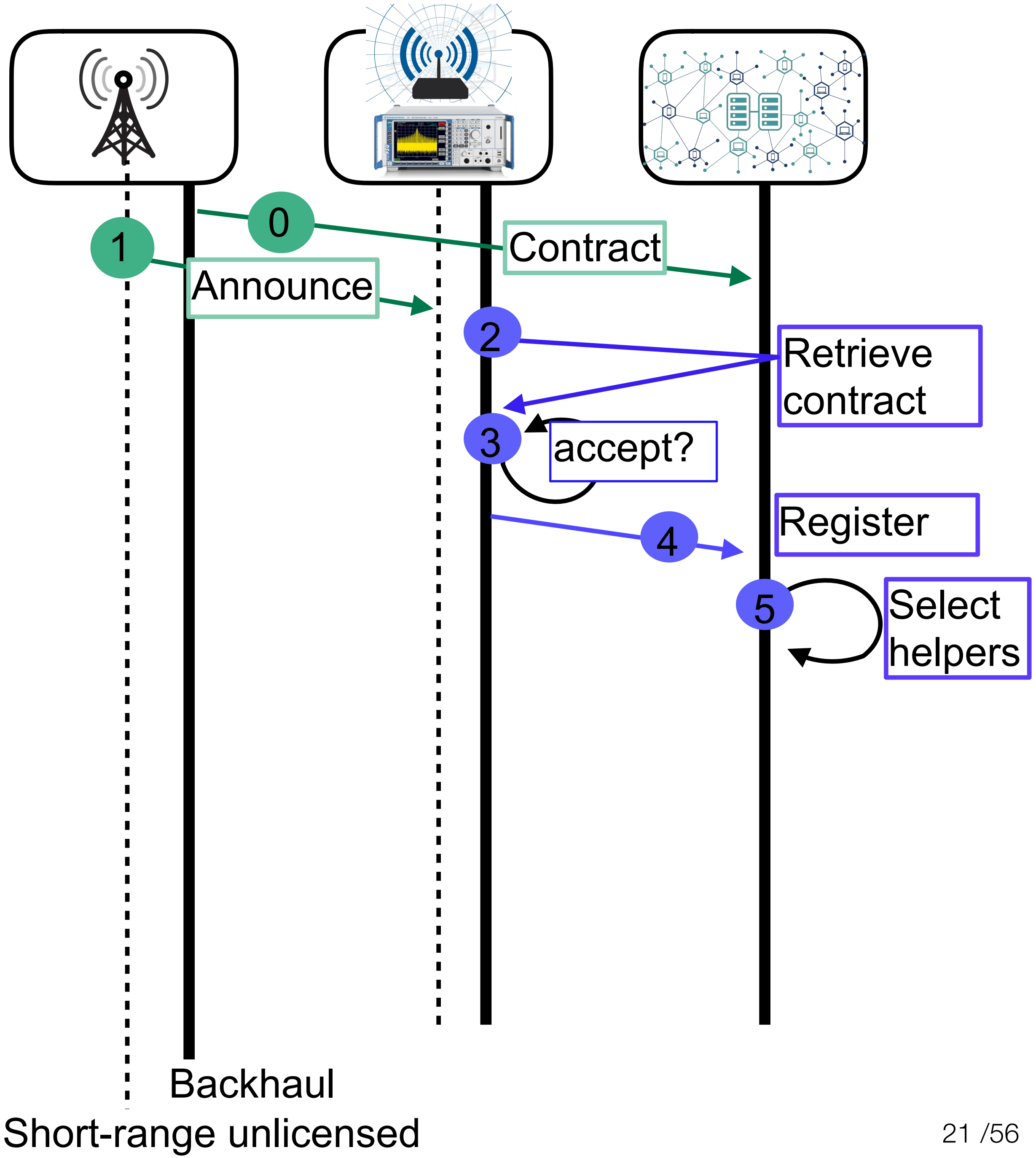
Message flow

- Helpers register:
  - accuracy in PU detection, false alarms
  - Sensing service price, location, etc.
- Helper selection:
  - Goal: SUN's profit is maximised
  - Input parameters: cost, sensing reliability, reputation etc.
  - Global sensing accuracy fulfils regulatory requirements

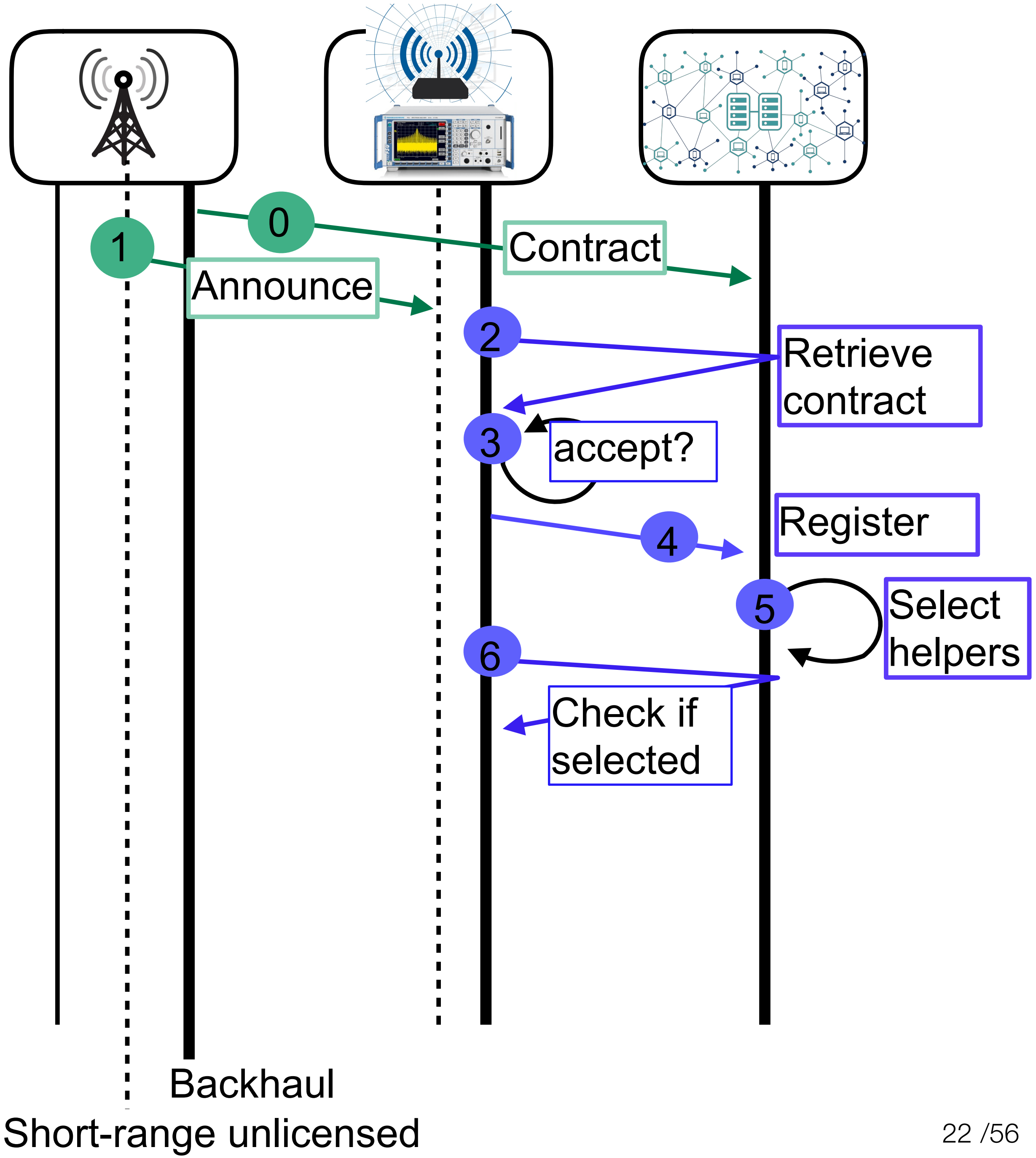


Message flow

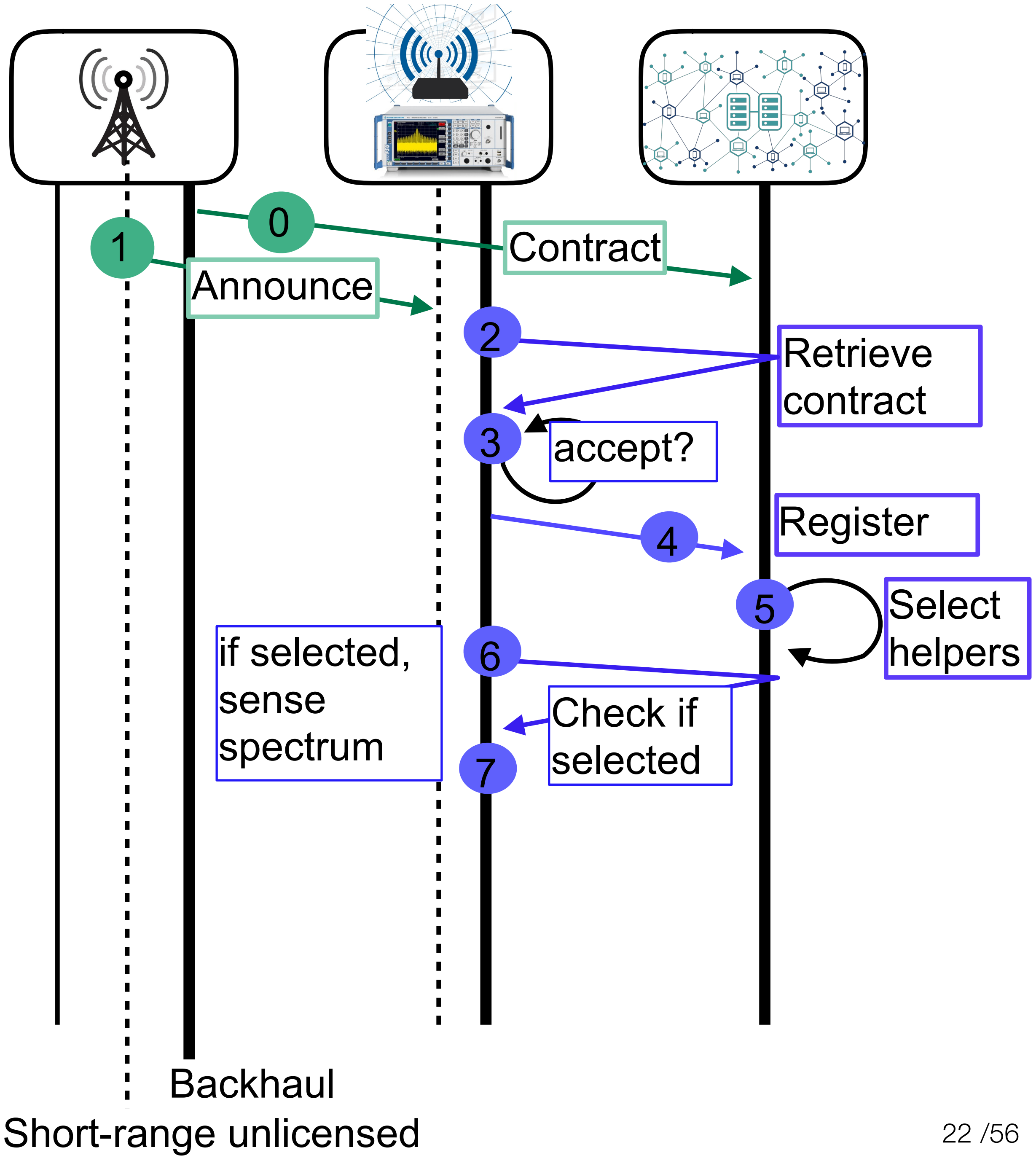
- Helpers register:
  - accuracy in PU detection, false alarms
  - Sensing service price, location, etc.
- Helper selection:
  - Goal: SUN’s profit is maximised
  - Input parameters: cost, sensing reliability, reputation etc.
  - Global sensing accuracy fulfils regulatory requirements



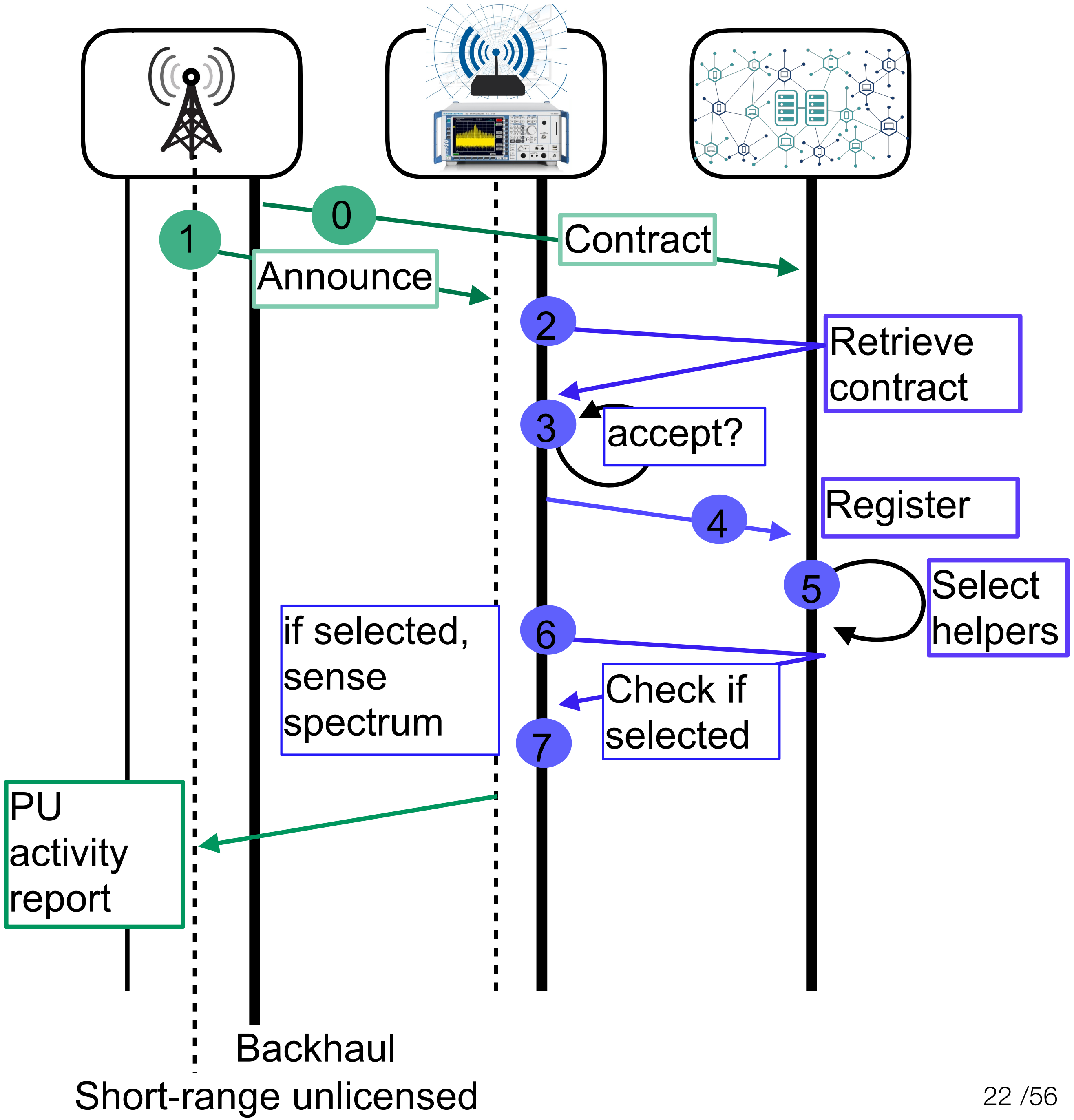
Message flow



Message flow

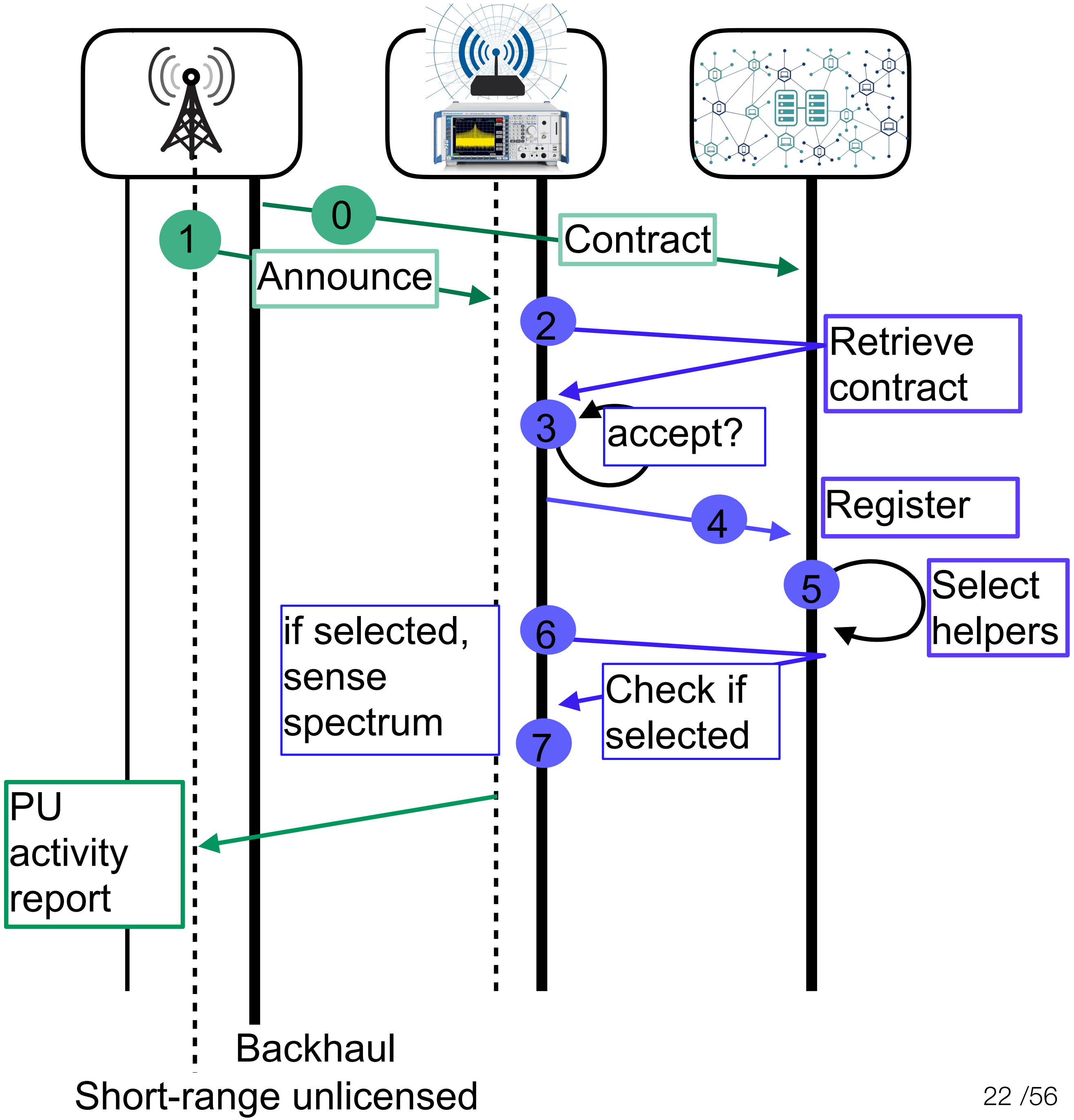


Message flow



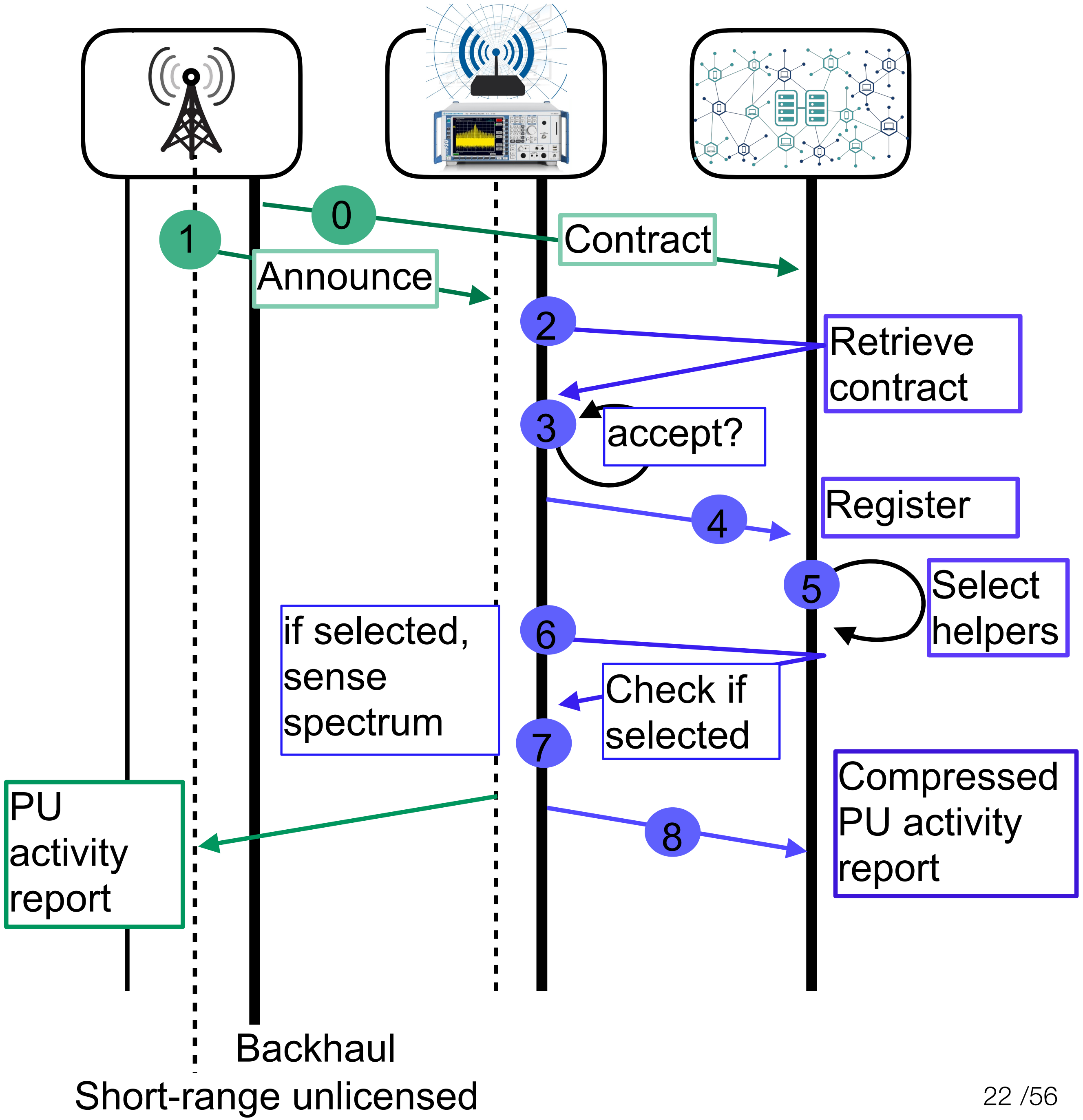
Message flow

- Spectrum sensing
  - Energy detection (hard decision: 0 or 1)
  - PU activity report (binary array)
  - SUN sensing decision fusion: majority voting
- Helper accuracy verification
  - Compressed reports
  - Malicious helper detection and payment
  - Goal: High malicious helper detection accuracy, low # blacklisted honest helpers
  - Payment to only whitelisted-helpers



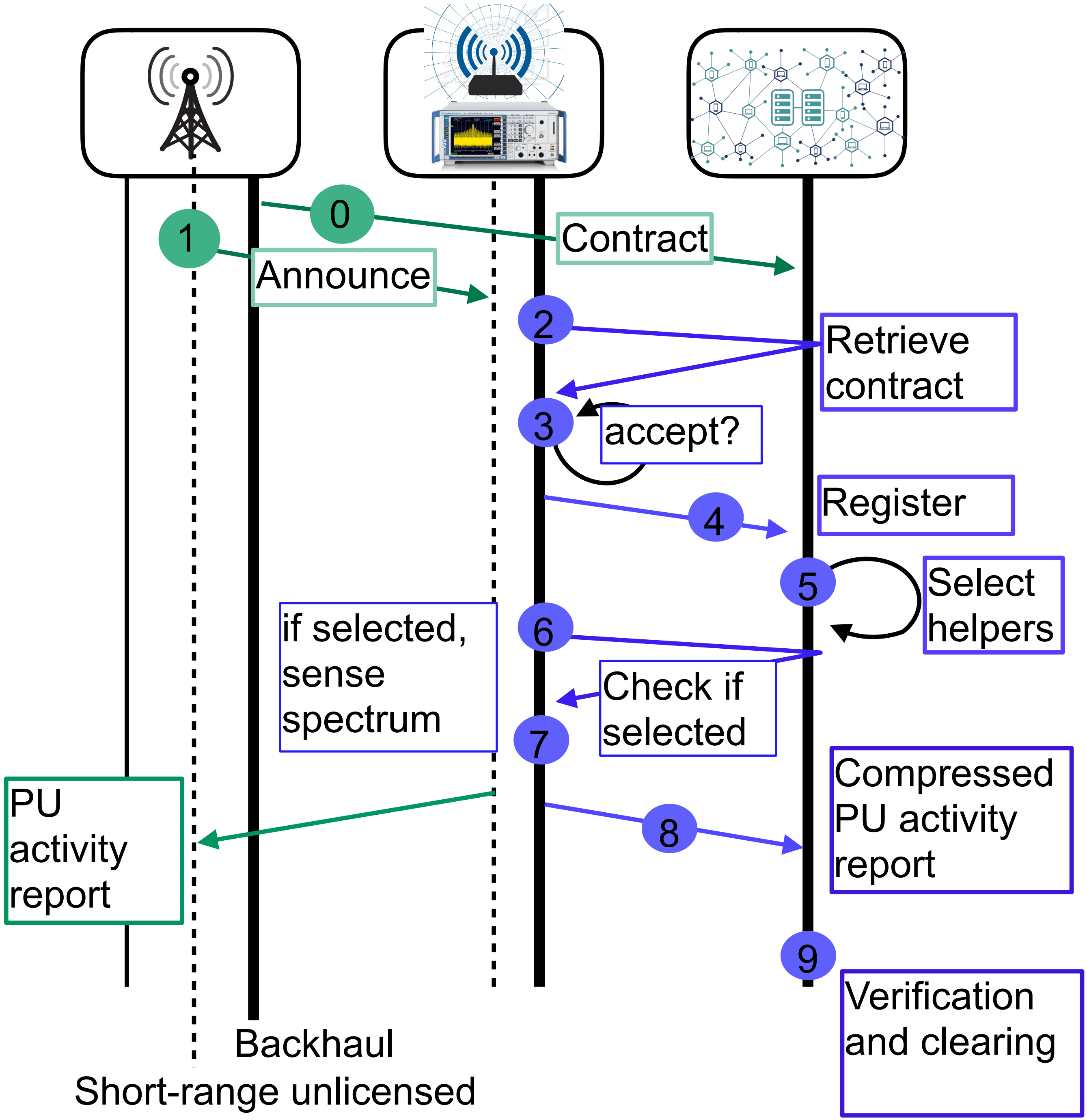
Message flow

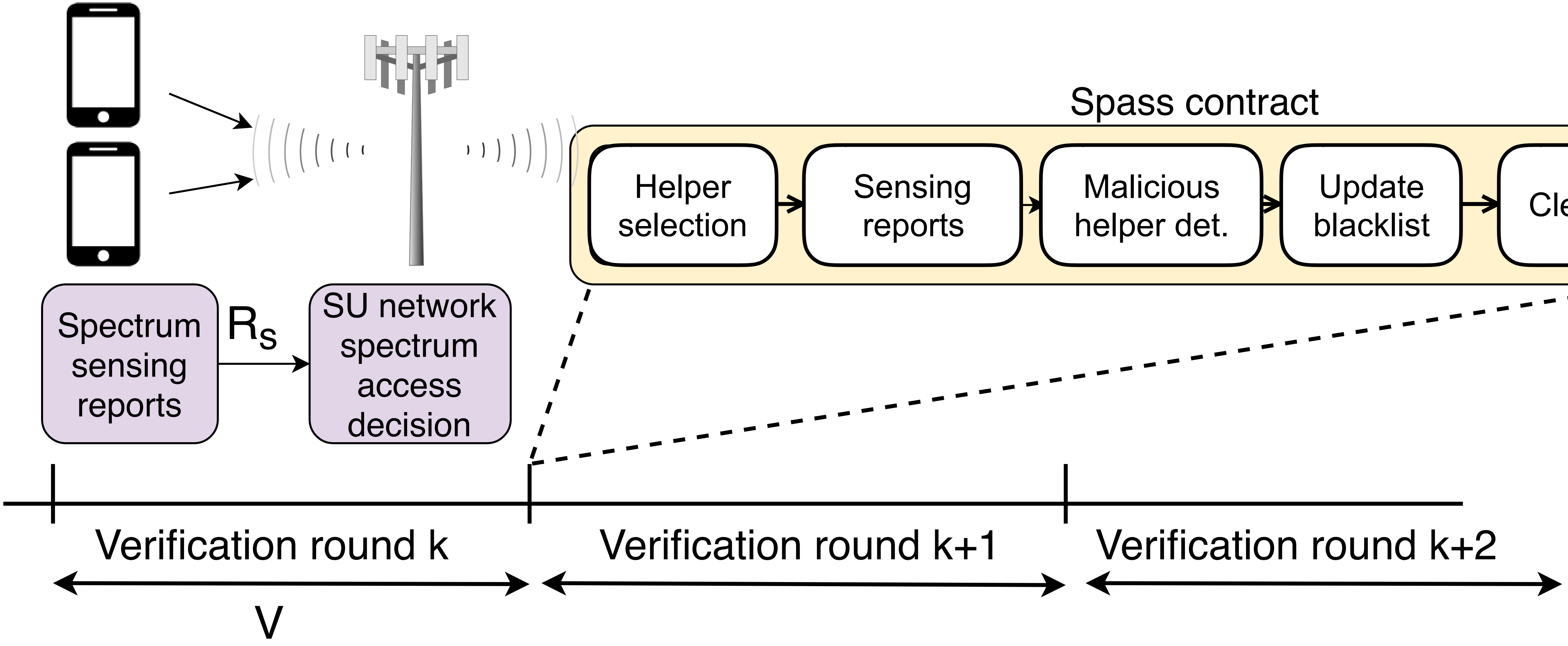
- Spectrum sensing
  - Energy detection (hard decision: 0 or 1)
  - PU activity report (binary array)
  - SUN sensing decision fusion: majority voting
- Helper accuracy verification
  - Compressed reports
  - Malicious helper detection and payment
  - Goal: High malicious helper detection accuracy, low # blacklisted honest helpers
  - Payment to only whitelisted-helpers



Message flow

- Spectrum sensing
  - Energy detection (hard decision: 0 or 1)
  - PU activity report (binary array)
  - SUN sensing decision fusion: majority voting
- Helper accuracy verification
  - Compressed reports
  - Malicious helper detection and payment
  - Goal: High malicious helper detection accuracy, low # blacklisted honest helpers
  - Payment to only whitelisted-helpers




Spass contract
Helper selection
Sensing reports
Malicious helper det.
Update blacklist
Clearing
Spectrum sensing reports
R_s
SU network spectrum access decision
Verification round k
Verification round k+1
Verification round k+2
V

How to *design* a smart contract so that SUN maximises its profit?

Spass

- How many helpers
- Verification round duration
- Cost of smart contracts
- Profit model

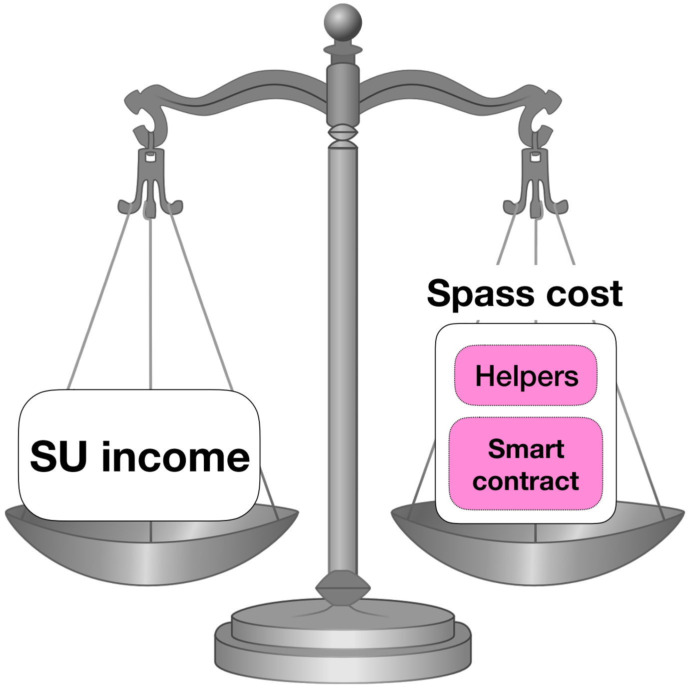
- Design goals
- Contract functionality
- **Optimal contract parameters**
- Malicious helper identification
- Performance
- Business feasibility of Spass

Assumptions

- Primary User (PU)
    - active with probability p_1 and inactive with p_0
    - known by SUN and helpers
- Homogenous helpers with identical sensing cost and accuracy
    - Probability of detection (P_d)
    - Probability of false alarm (P_f)
- Malicious helpers with identical sensing accuracy: *free riders (no energy consumption for sensing or colluding)*
    - do not sense the spectrum but generate data using their statistical knowledge of p_0
    - Fraction of malicious helper population known by the SUN
    - Malicious helpers do not collude and do not change their strategy

SUN profit

- **Income**: Monetary gain accumulated over all verification rounds by serving its customers over the opportunistic spectrum
- **Payment**: Helpers sensing service and the smart contract cost



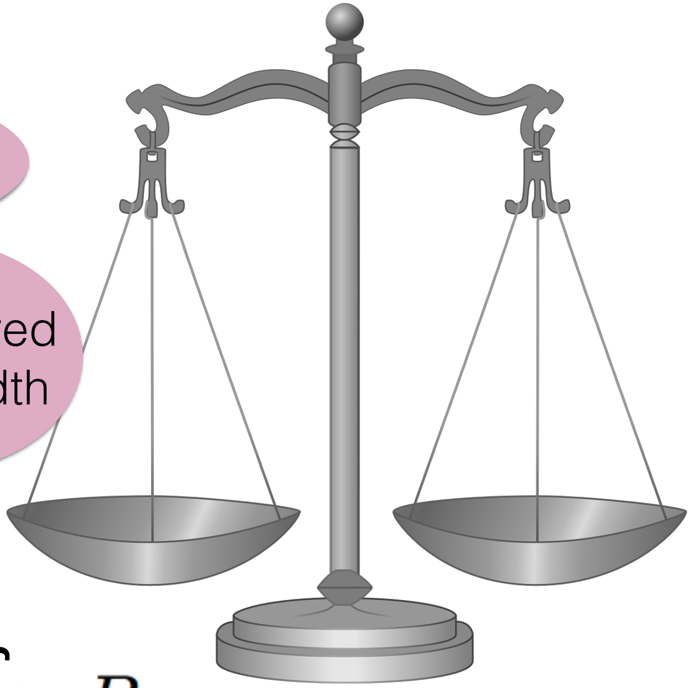
SUN profit in a verification round

SUN profit in a verification round

Technology dependent spectral efficiency

client pays μ euros/bit for second

Discovered Bandwidth



$$\Upsilon^{+} = \mu \times \kappa \times \mathcal{U}^{\mathrm{Spass}} \times B$$

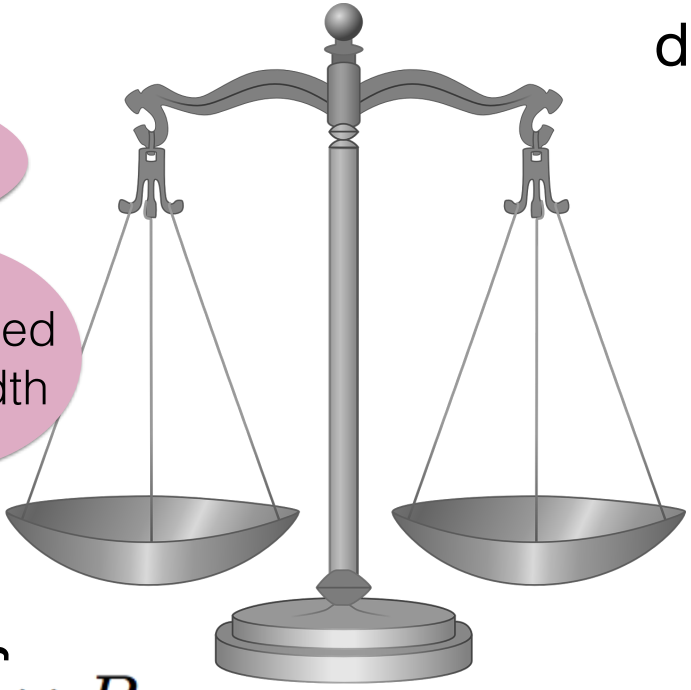
SUN profit in a verification round

Assumption: write operation is the dominant cost in Ethereum usage

Technology dependent spectral efficiency

client pays μ euros/bit for second

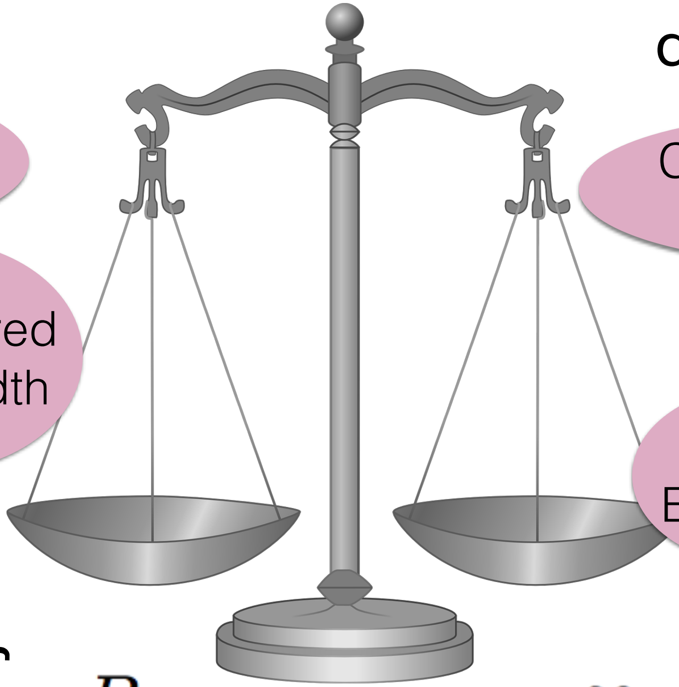
Discovered Bandwidth



$$\Upsilon^+ = \mu \times \kappa \times \mathcal{U}^{\text{Spass}} \times B$$

SUN profit in a verification round

Assumption: write operation is the dominant cost in Ethereum usage



Technology dependent spectral efficiency

client pays μ euros/bit for second

Discovered Bandwidth

$$\Upsilon^{+} = \mu \times \kappa \times \mathcal{U}^{\mathrm{Spass}} \times B$$

Compression ratio

rate of sensing

cost of sensing

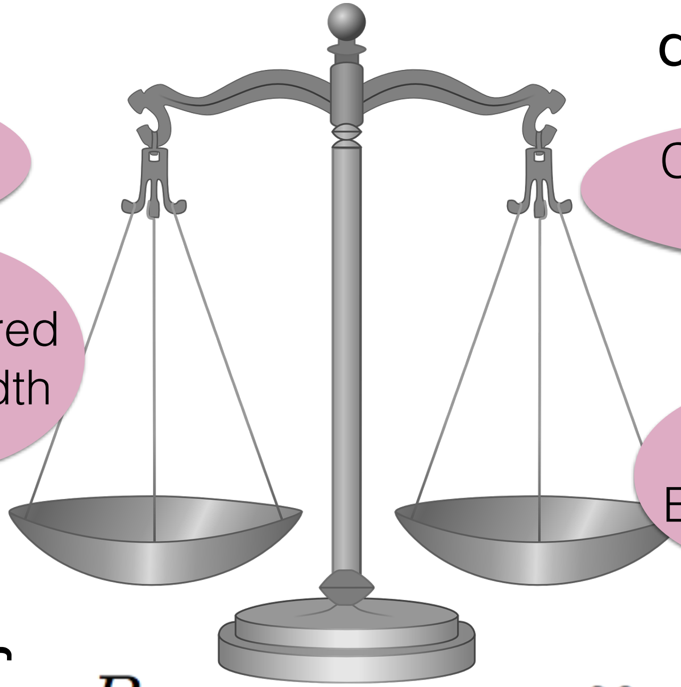
cost of Ethereum

number of helpers

$$\Upsilon^{-} = (\mu_{eth}\frac{R_s}{\beta} + \mu_s R_s) \times H$$

SUN profit in a verification round

Assumption: write operation is the dominant cost in Ethereum usage



Technology dependent spectral efficiency

client pays μ euros/bit for second

Discovered Bandwidth

Compression ratio

rate of sensing

cost of sensing

cost of Ethereum

number of helpers

$$\Upsilon^+ = \mu \times \kappa \times \mathcal{U}^{\overbrace{\text{Spass}}} \times B$$

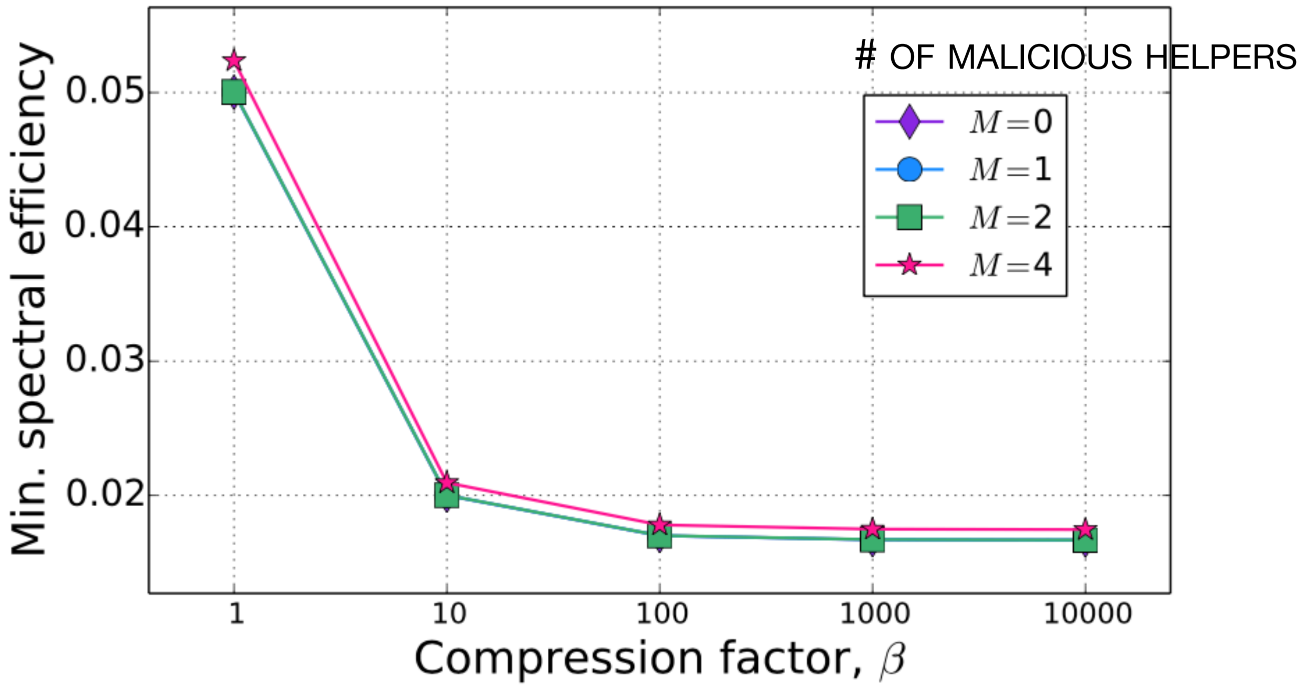
$$\Upsilon^- = (\mu_{eth} \frac{R_s}{\beta} + \mu_s R_s) \times H$$

We find the operation range of an SU network in which it can make profit from Spass: net profit>0 $\Delta\Upsilon = \Upsilon^+ - \Upsilon^- > 0$

When is Spass profitable?

Required min. spectral efficiency

$$\kappa > \frac{R_s(\mu_{eth}/\beta + \mu_s) \times H}{\mu \times \mathcal{U}^{\text{Spass}} \times B}$$



Impact of compression β under $H = 4$.

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s (\frac{\mu_{eth}}{\beta} + \mu_s) \right) \tag{8}$$

$$N_V = T_c / V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \tag{12}$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f* \text{ and } q_d \geqslant q_d* \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f* \text{ and } \bar{p_d}(H) \geqslant p_d* \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity

$$\mathbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H-\tilde{M}_v) R_s (\frac{\mu_{eth}}{\beta} + \mu_s) \right) \tag{8}$$

$$N_V = T_c/V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \tag{12}$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa \mathcal{U}_v - (H - \tilde{M}_v) R_s \left(\frac{\mu_{eth}}{\beta} + \mu_s\right) \right) \quad (8)$$

$$N_V = T_c / V \quad (9)$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \quad (10)$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \quad (11)$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \quad (12)$$

$$\binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}.$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \quad (13)$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \quad (14)$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \quad (15)$$

$$S = \frac{R_s V}{\beta} \quad (16)$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \quad (17)$$

- Utility: probability of detecting the spectrum opportunity

Expected utility if H helpers sense

Expected false alarm probability

False alarm probability under j malicious helpers and total H helpers

$$\mathbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa \mathcal{U}_v - (H - \tilde{M}_v) R_s \left(\frac{\mu_{eth}}{\beta} + \mu_s \right) \right) \quad (8)$$

$$N_V = T_c / V \quad (9)$$

$$\mathcal{U}_v = p_0 (1 - \bar{p_f}(H)) \quad (10)$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \quad (11)$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \quad (12)$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \quad (13)$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \quad (14)$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \quad (15)$$

$$S = \frac{R_s V}{\beta} \quad (16)$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \quad (17)$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s (\frac{\mu_{eth}}{\beta} + \mu_s) \right) \tag{8}$$

$$N_V = T_c / V \tag{9}$$

$$\mathcal{U}_v = p_0 (1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \tag{12}$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Helpers *detected as malicious* are not paid (false and true detection)

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s (\frac{\mu_{eth}}{\beta} + \mu_s) \right) \quad (8)$$

$$N_V = T_c / V \quad (9)$$

$$\mathcal{U}_v = p_0 (1 - \bar{p_f}(H)) \quad (10)$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \quad (11)$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \quad (12)$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \quad (13)$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \quad (14)$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \quad (15)$$

$$S = \frac{R_s V}{\beta} \quad (16)$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \quad (17)$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Helpers *detected as malicious* are not paid (false and true detection)

Fraction of malicious helpers in all population

q_d, q_f: true and false detections

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s \left(\frac{\mu_{eth}}{\beta} + \mu_s\right) \right) \tag{8}$$

$$N_V = T_c/V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \tag{12}$$

$$\binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}.$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f* \text{ and } q_d \geqslant q_d* \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f* \text{ and } \bar{p_d}(H) \geqslant p_d* \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Malicious helpers are not paid
- For sustainable operation, honest helpers are not blacklisted and malicious helpers are detected with high probability

$$\mathbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s \left(\frac{\mu_{eth}}{\beta} + \mu_s\right) \right) \tag{8}$$

$$N_V = T_c / V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \tag{12}$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Malicious helpers are not paid
- For sustainable operation, honest helpers are not blacklisted and malicious helpers are detected with high probability
- Sensing report size under compression factor ß

$$\textbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s \left(\frac{\mu_{eth}}{\beta} + \mu_s\right) \right) \tag{8}$$

$$N_V = T_c / V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H, j) \tag{11}$$

$$p_f(H, j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1 - p_f^m)^{j-i} \tag{12}$$

$$\binom{H-j}{K-i} (p_f^h)^{K-i} (1 - p_f^h)^{H-j-K+i}.$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \tag{15}$$

$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Malicious helpers are not paid
- For sustainable operation, honest helpers are not blacklisted and malicious helpers are detected with high probability
- Sensing report size under compression factor ß

R_s : Rate of sensing (bps)
ß: compression factor

$$\mathbf{P1:} \max_{V,H} \left(V \sum_{v=1}^{N_V} \mu\kappa\mathcal{U}_v - (H - \tilde{M}_v) R_s (\frac{\mu_{eth}}{\beta} + \mu_s) \right) \tag{8}$$

$$N_V = T_c / V \tag{9}$$

$$\mathcal{U}_v = p_0(1 - \bar{p_f}(H)) \tag{10}$$

$$\bar{p_f}(H) = \sum_{j=0}^{H} \binom{H}{j} \psi^j (1-\psi)^{H-j} p_f(H,j) \tag{11}$$

$$p_f(H,j) = \sum_{K-\lceil H/2 \rceil}^{H} \sum_{i=0}^{\min(H,j)} \binom{j}{i} (p_f^m)^i (1-p_f^m)^{j-i} \binom{H-j}{K-i} (p_f^h)^{K-i} (1-p_f^h)^{H-j-K+i}. \tag{12}$$

$$\tilde{M}_v = H(\psi q_d + (1-\psi) q_f) \tag{13}$$

$$q_f \leqslant q_f * \text{ and } q_d \geqslant q_d * \tag{14}$$

$$\bar{p_f}(H) \leqslant p_f * \text{ and } \bar{p_d}(H) \geqslant p_d * \tag{15}$$

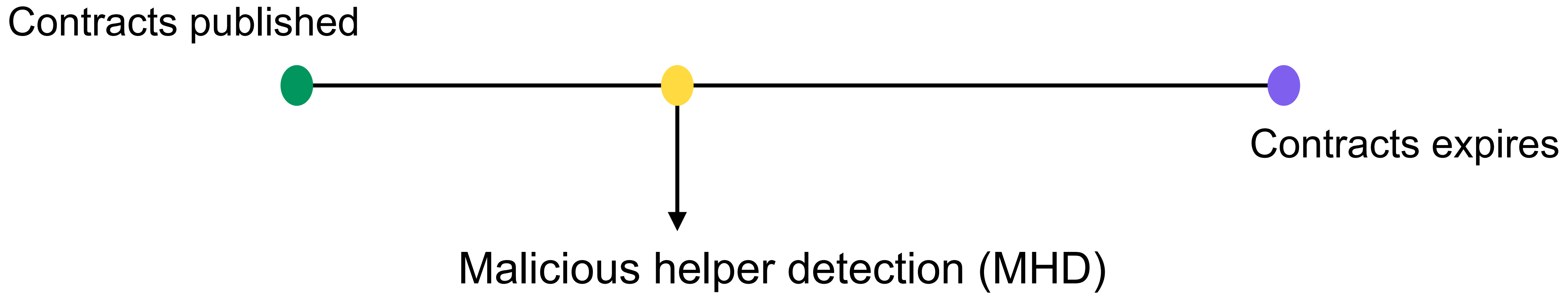
$$S = \frac{R_s V}{\beta} \tag{16}$$

$$V \in \{1, \cdots, T_c\} \text{ and } H \in \{1, \cdots, N\} \tag{17}$$

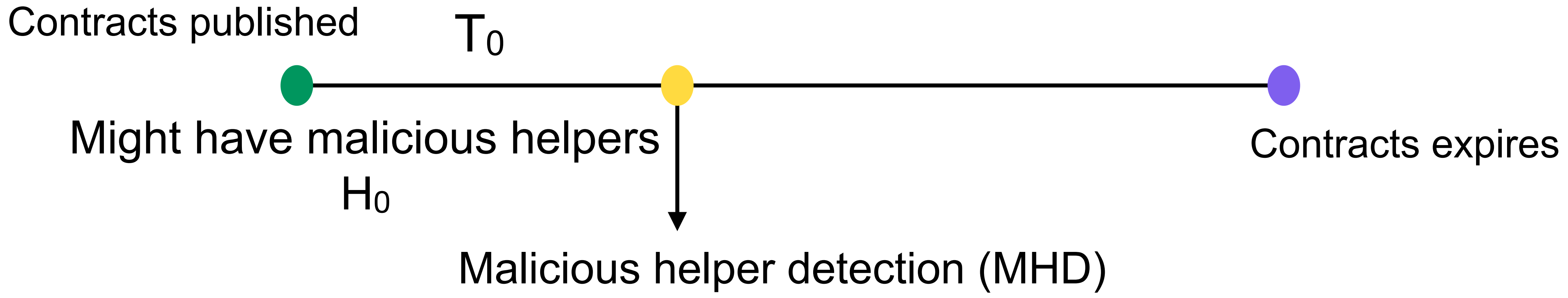
- Utility: probability of detecting the spectrum opportunity
- Regulatory requirements on sensing accuracy are satisfied
- Malicious helpers are not paid
- For sustainable operation, honest helpers are not blacklisted and malicious helpers are detected with high probability
- Sensing report size under compression factor beta

But, we do not have a closed formula for malicious helper detection accuracy!

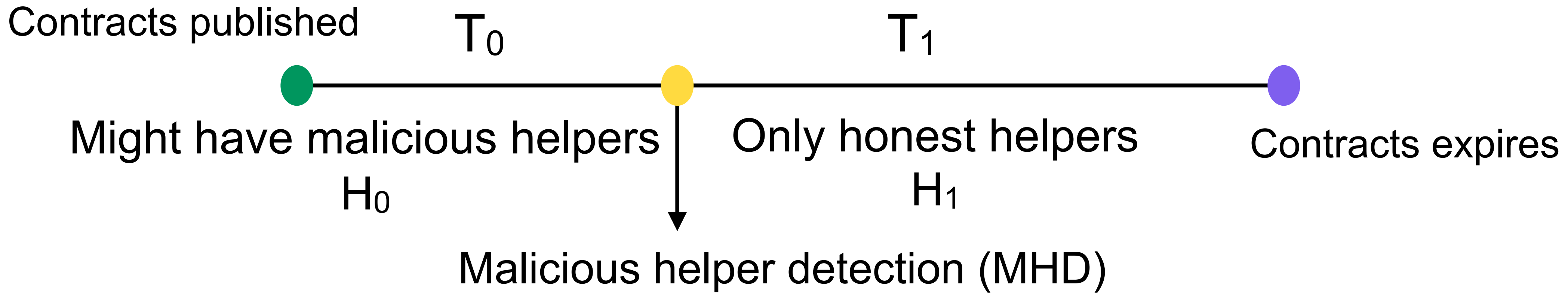
Simplified problem: two phase operation



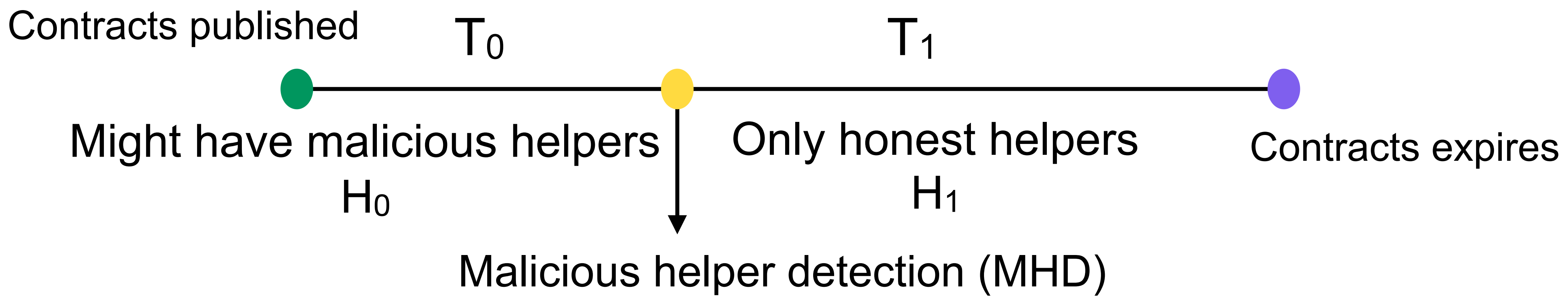
Simplified problem: two phase operation



Simplified problem: two phase operation



Simplified problem: two phase operation



- Requirement: After the first phase, all malicious helpers are (must be) detected by our MHD algorithm
- Malicious helpers do not change their strategy

Simplified problem: two phase operation

- Now, decision on H_0, H_1, T_0
  - $T_{0:}$ the minimum duration that the MHD algorithm needs for detection of all malicious helpers
  - We find T_0 via Monte Carlo simulations
  - H_0, $H_{1:}$ exhaustive search $O(N^2)$ where N is number of candidates

$$\textbf{P2:} \max_{T_0,H_0,H_1} T_0(\mu\kappa\mathcal{U}_0 - H_0R_s(\frac{\mu_{eth}}{\beta}+\mu_s)) \\ + (T_c - T_0)(\mu\kappa\mathcal{U}_1 - H_1R_s(\frac{\mu_{eth}}{\beta}+\mu_s) \quad (18)$$

$$T_0 \leqslant T_c \quad (19)$$

$$\mathcal{U}_0 = p_0(1-\bar{p_f}(H_0)) \quad (20)$$

$$\mathcal{U}_1 = p_0(1-p_f(H_1)) \quad (21)$$

$$H_0 \geqslant H_1 \quad (22)$$

$$p_f(H_1) \geqslant p_f^* \text{ and } p_d(H_1) \geqslant p_d^* \quad (23)$$

$$\bar{p_f}(H_0) \geqslant p_f^* \text{ and } \bar{p_d}(H_0) \geqslant p_d^* \quad (24)$$

$$S_{\min} \geqslant \frac{R_sT_0}{\beta} \quad (25)$$

How to decrease cost of Spass?

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

$$\mathcal{X} = \sum_{i=1}^{L} q_i \log_2 \frac{1}{q_i}$$

q_i: probability of symbol i

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

$$\mathcal{X} = \sum_{i=1}^{L} q_i \log_2 \frac{1}{q_i}$$

q_i: probability of symbol i

Desirable! But, limited compression depending on q_i (occupancy state of the channel, 0 or 1)

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

$$\mathcal{X} = \sum_{i=1}^{L} q_i \log_2 \frac{1}{q_i}$$

q_i: probability of symbol i

Desirable! But, limited compression depending on q_i (occupancy state of the channel, 0 or 1)

Lossy compression
(removing bits randomly)

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

$$\mathcal{X} = \sum_{i=1}^{L} q_i \log_2 \frac{1}{q_i}$$

q_i: probability of symbol i

Desirable! But, limited compression depending on q_i (occupancy state of the channel, 0 or 1)

Lossy compression
(removing bits randomly)

Compression ratio: β

How to decrease cost of Spass?

Our approach: Decrease the amount of data written to the contract

Lossless compression
(variable length codes)

$$\mathcal{X} = \sum_{i=1}^{L} q_i \log_2 \frac{1}{q_i}$$

q_i: probability of symbol i

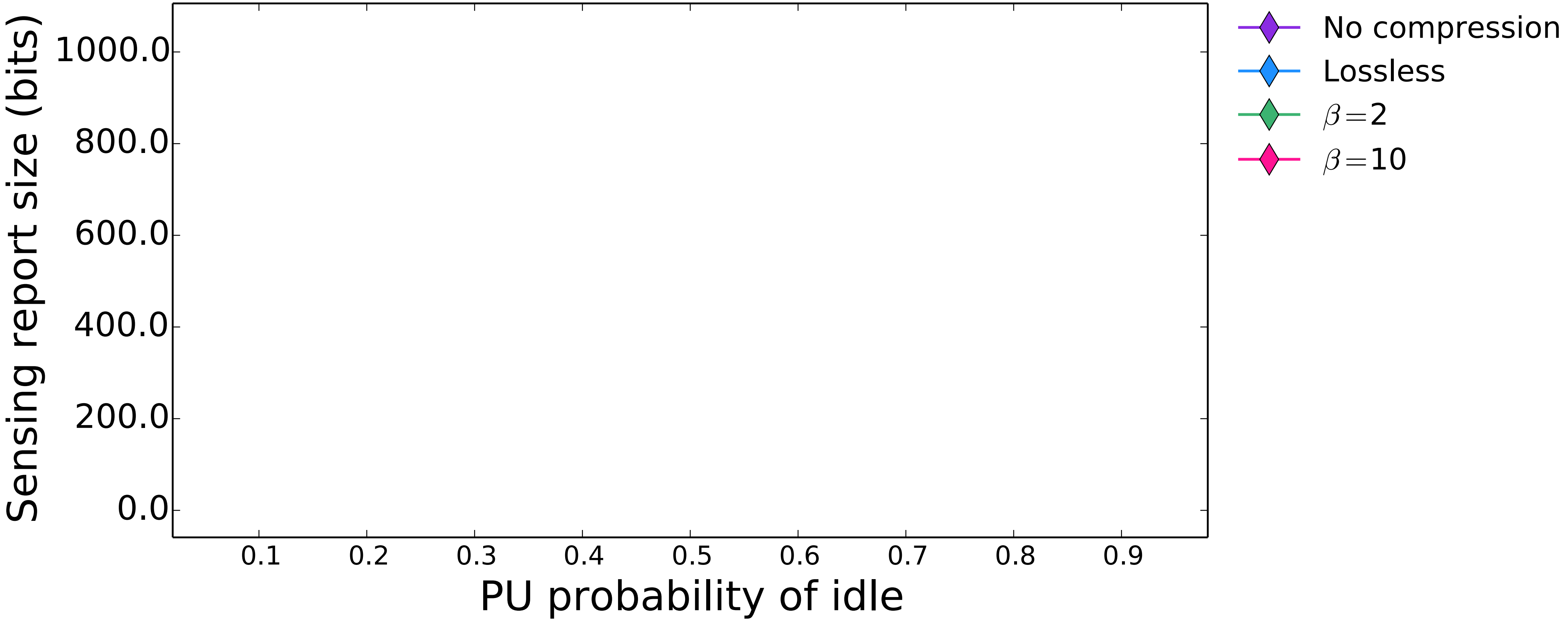
Desirable! But, limited compression depending on q_i (occupancy state of the channel, 0 or 1)

Lossy compression
(removing bits randomly)

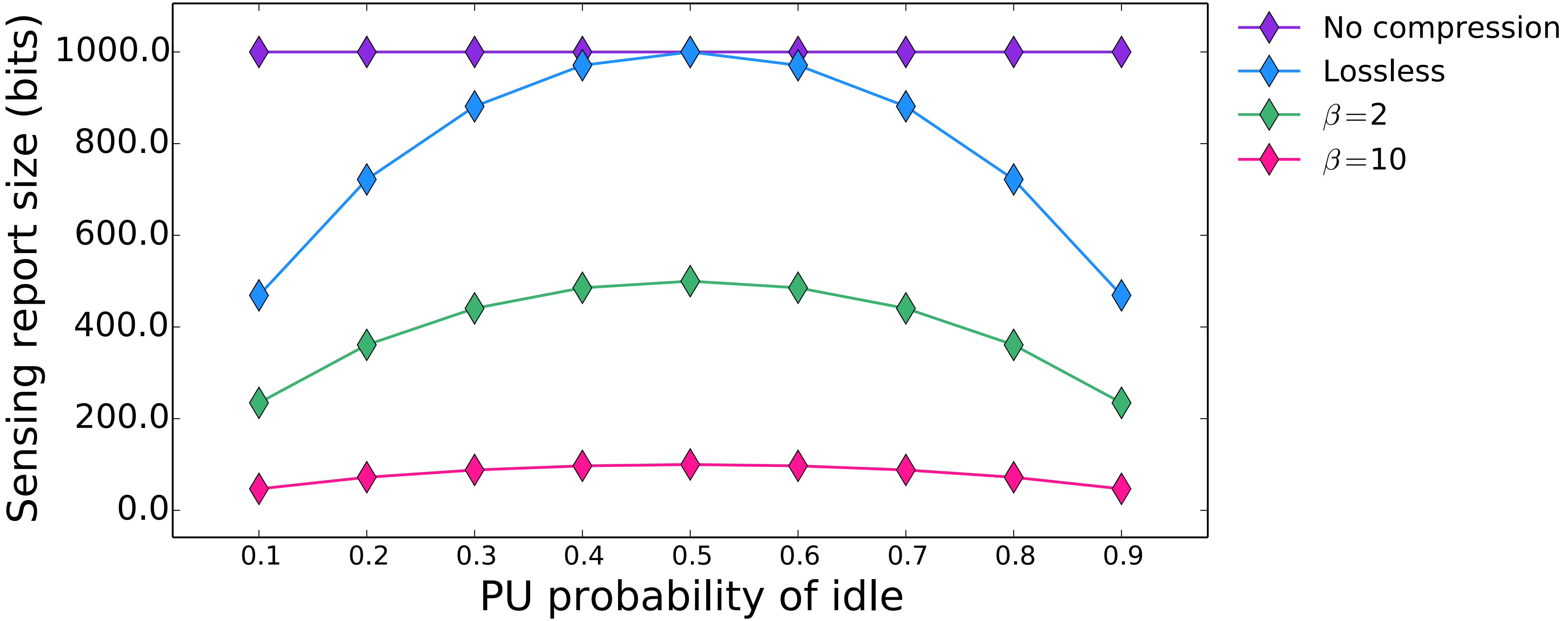
Compression ratio: β

Might affect malicious helper detection algorithm's accuracy

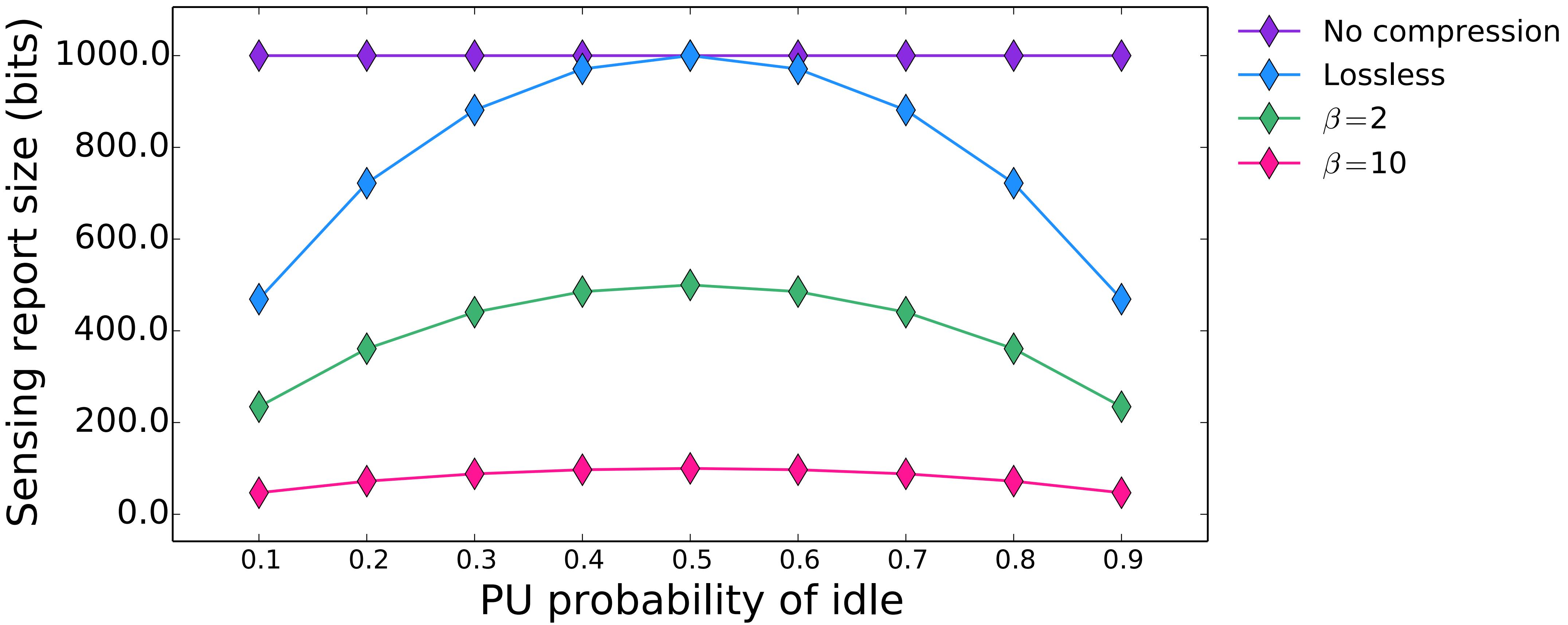
How to decrease cost of Spass?



How to decrease cost of Spass?



How to decrease cost of Spass?



- Higher compression when low/high PU idle

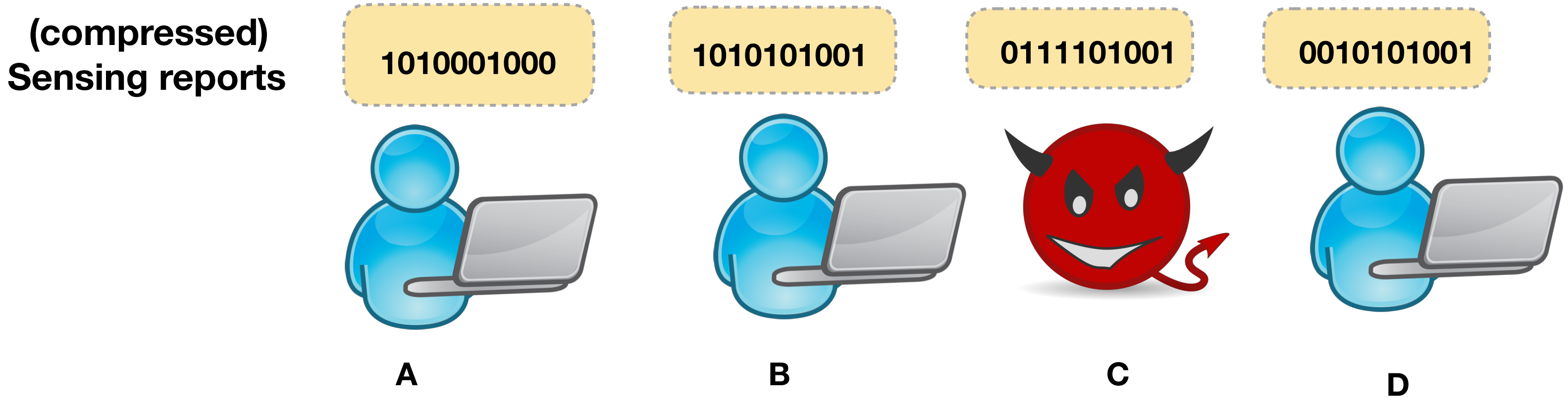
Spass

- Design goals
- Contract functionality
- Optimal contract parameters
- Malicious helper identification
- Performance
- Business feasibility of Spass

Spass

- Design goals
- Contract functionality
- Optimal contract parameters
- **Malicious helper identification**
- Performance
- Business feasibility of Spass

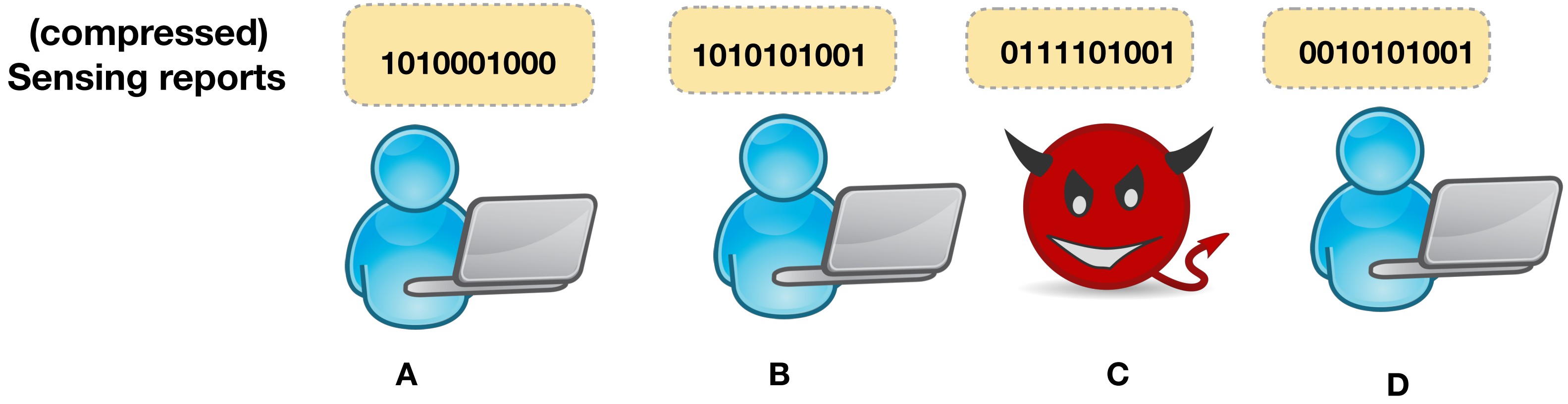
Clustering-based malicious Helper Identification (CHI)



Honest helper model:
Identical sensing accuracy
P^h_d, P^h_{fa}

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m_d = (1-p_0)^2$.

$\mathbf{C}$lustering-based malicious $\mathbf{H}$elper $\mathbf{I}$dentification (CHI)

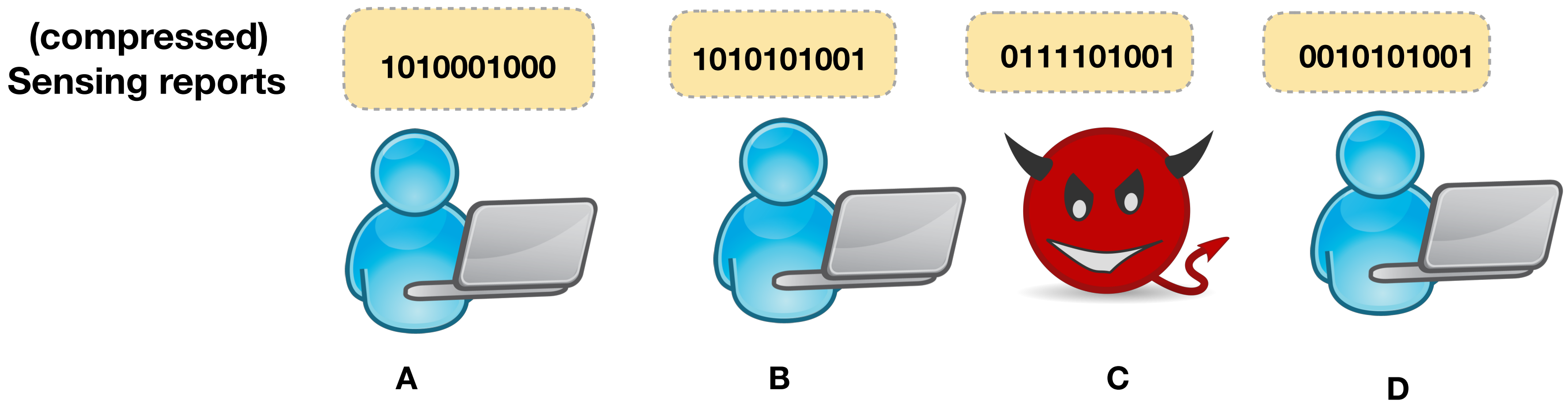


- **Distance between two helpers**
    - normalized Hamming distance of two helper reports

Honest helper model:
Identical sensing accuracy
P^h_d, P^h_{fa}

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m_d = (1-p_0)^2$.

Clustering-based malicious **H**elper **I**dentification (**CHI**)

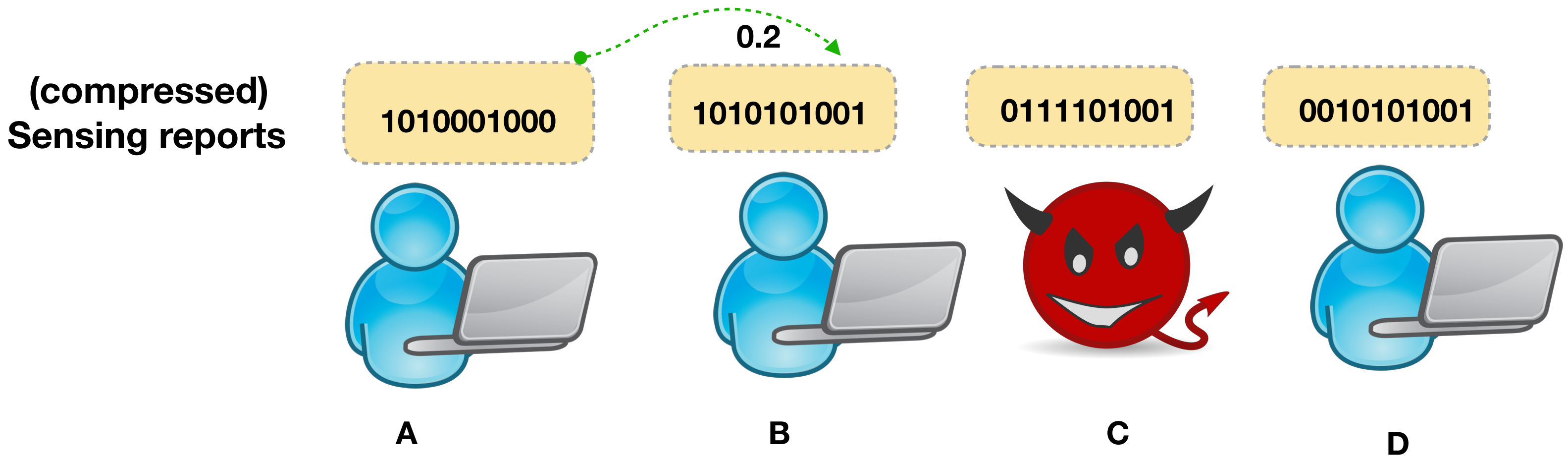


- **Distance between two helpers**
  - normalized Hamming distance of two helper reports

Honest helper model:
Identical sensing accuracy
P^h_d, P^h_{fa}

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m_d = (1-p_0)^2$.

Clustering-based malicious Helper Identification (CHI)

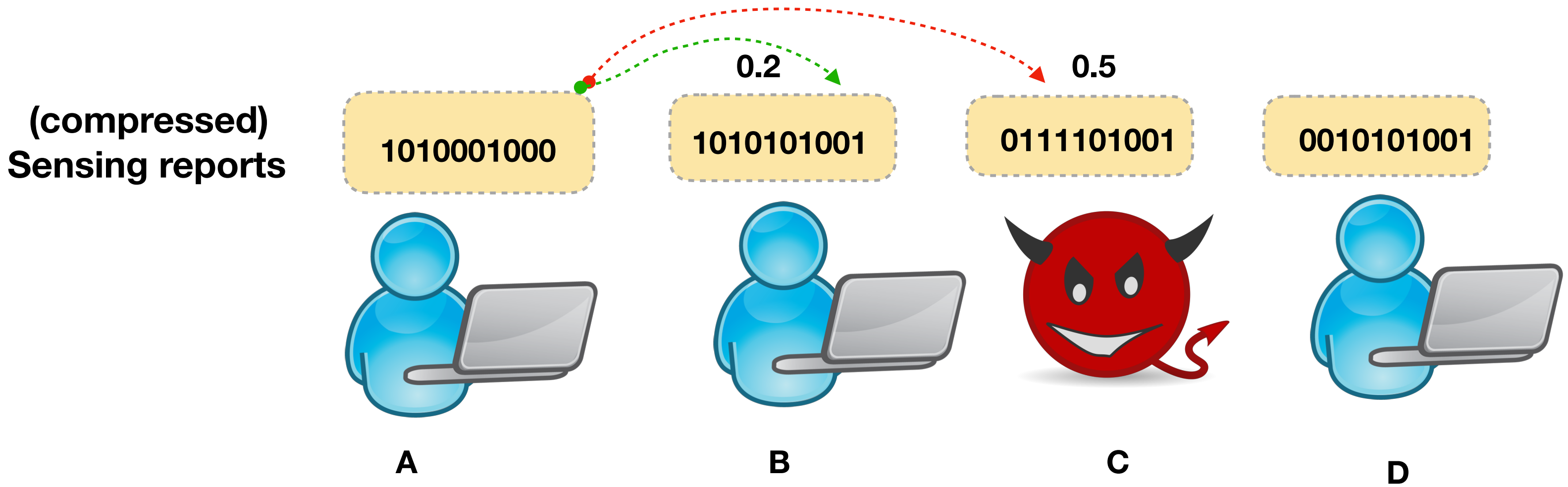


- **Distance between two helpers**
  - normalized Hamming distance of two helper reports

Honest helper model:
Identical sensing accuracy
$P^h{}_d$, $P^h{}_{fa}$

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m{}_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m{}_d = (1-p_0)^2$.

Clustering-based malicious Helper Identification (CHI)

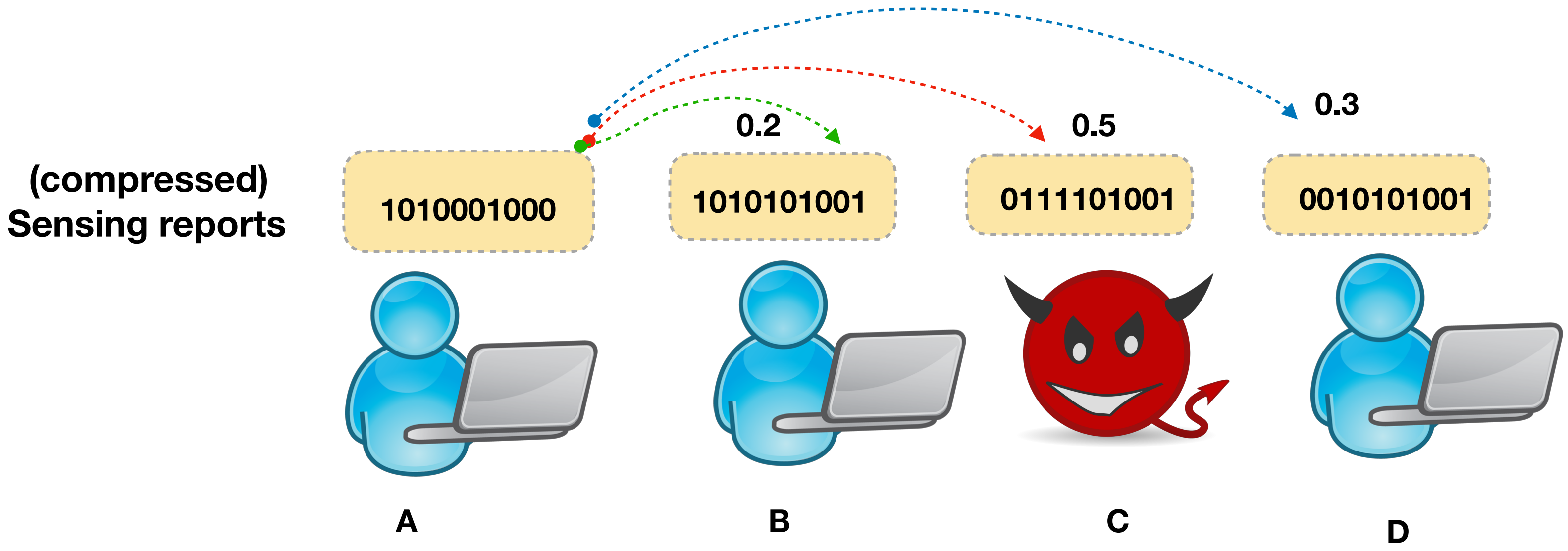


- **Distance between two helpers**
  - normalized Hamming distance of two helper reports

Honest helper model:
Identical sensing accuracy
P^h_d, P^h_{fa}

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m_d = (1-p_0)^2$.

Clustering-based malicious Helper Identification (CHI)

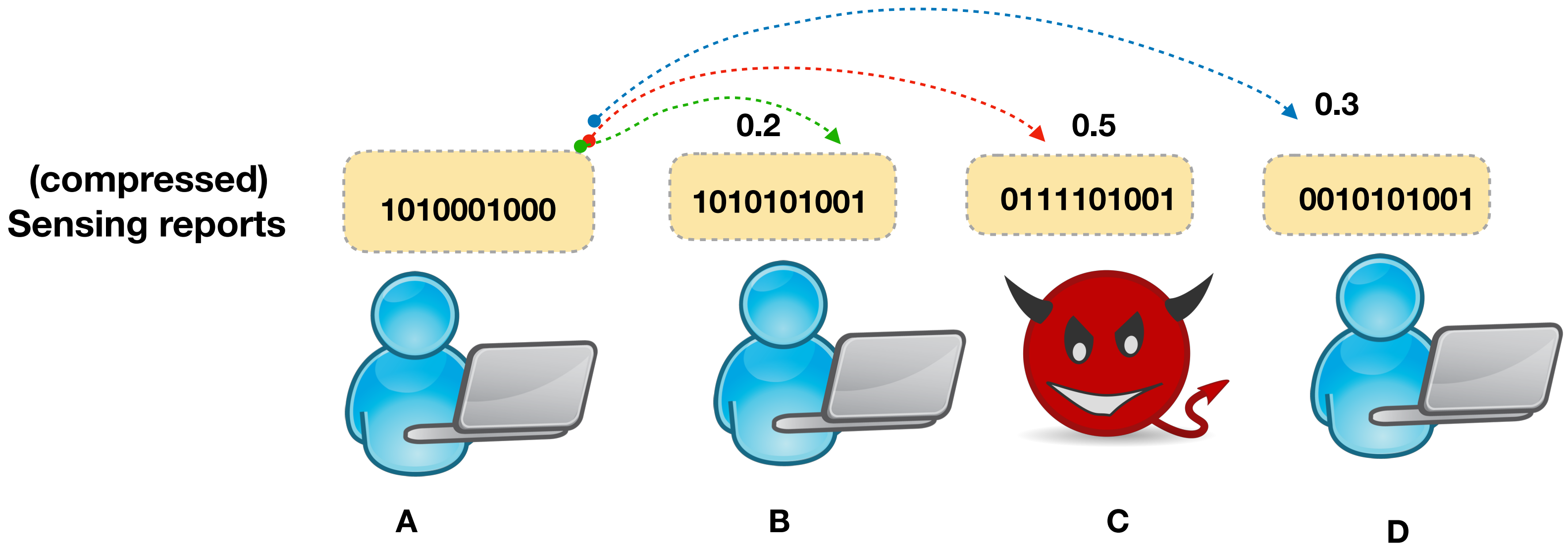


- **Distance between two helpers**
  - normalized Hamming distance of two helper reports

Honest helper model:
Identical sensing accuracy
$P^h{}_d$, $P^h{}_{fa}$

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m{}_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m{}_d = (1-p_0)^2$.

Clustering-based malicious Helper Identification (CHI)

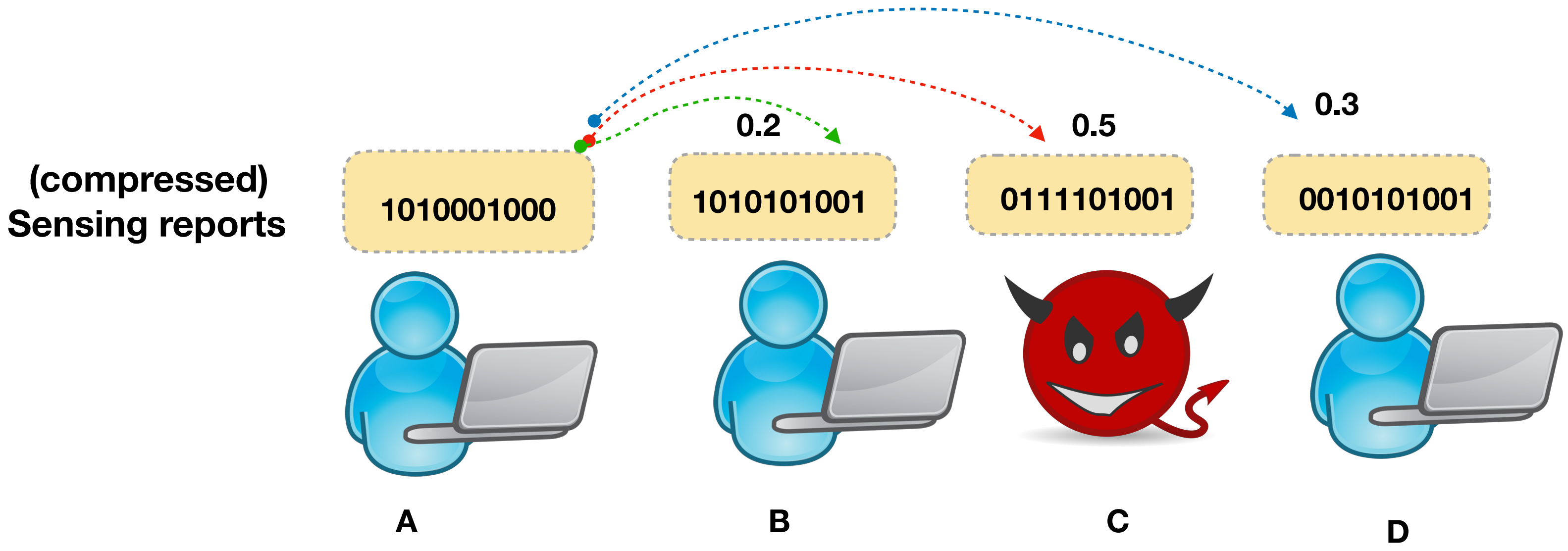


- **Distance between two helpers**
  - normalized Hamming distance of two helper reports
- **Helper score**: x-percentile of its distance from other helpers

Honest helper model:
Identical sensing accuracy
P^h_d, P^h_{fa}

Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m_d = (1-p_0)^2$.

Clustering-based malicious **H**elper **I**dentification (**CHI**)



Honest helper model:
Identical sensing accuracy
$P^h{}_d$, $P^h{}_{fa}$

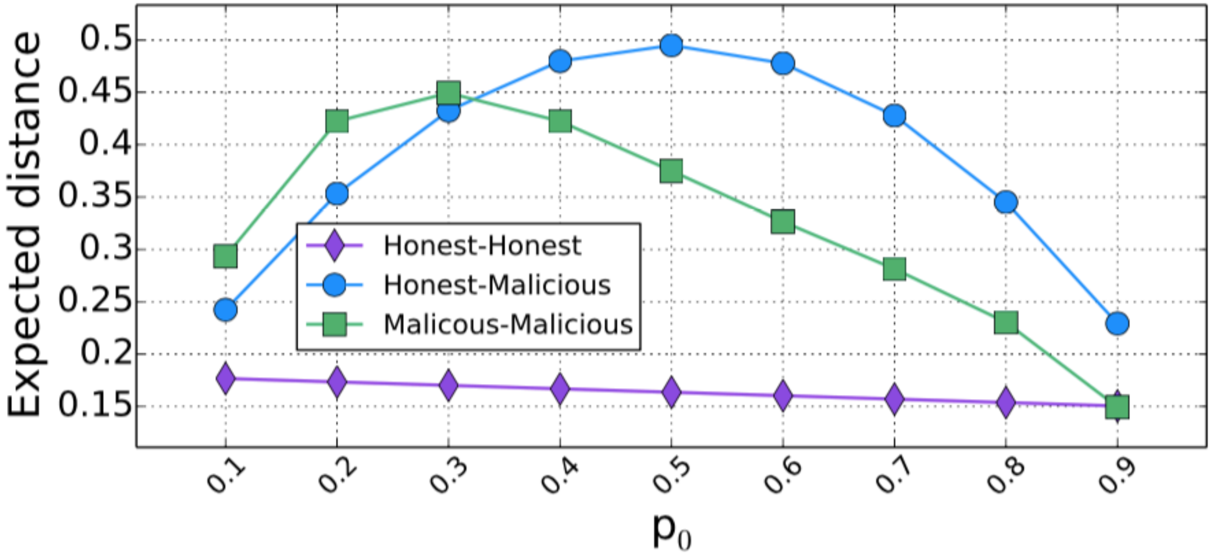
Malicious helper model:
Knows P_0,
Generates fake sensing bits without performing sensing
1 with prob $\alpha_1 = (1 - p_0)$.
$P^m{}_f = p_0\alpha_1 = p_0(1-p_0)$
$P^m{}_d = (1-p_0)^2$.

- **Distance between two helpers**
  - normalized Hamming distance of two helper reports
- **Helper score**: x-percentile of its distance from other helpers

If 50-percentile: Score-A = [0.2, 0.3, 0.5]

Key insight of CHI

- Honest helpers with similar sensing reports
    - Short distance from each other
    - Similar and low scores
- Malicious helpers (do not collude and do not sense the spectrum)
    - High distance from each other and from honest helpers
- Two clusters
    - Honest helper cluster and malicious helper cluster



(b) Expected distance between two helpers according to their type with increasing p_0.

CHI: one or two clusters?

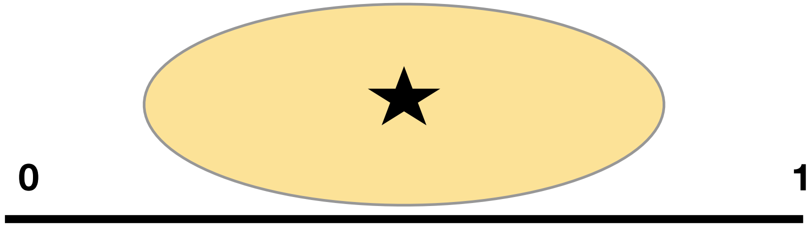
CHI: one or two clusters?

0 1

Helper scores

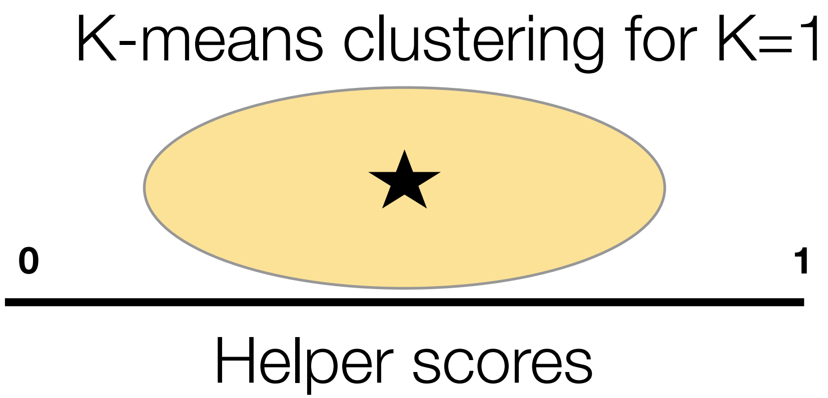
CHI: one or two clusters?

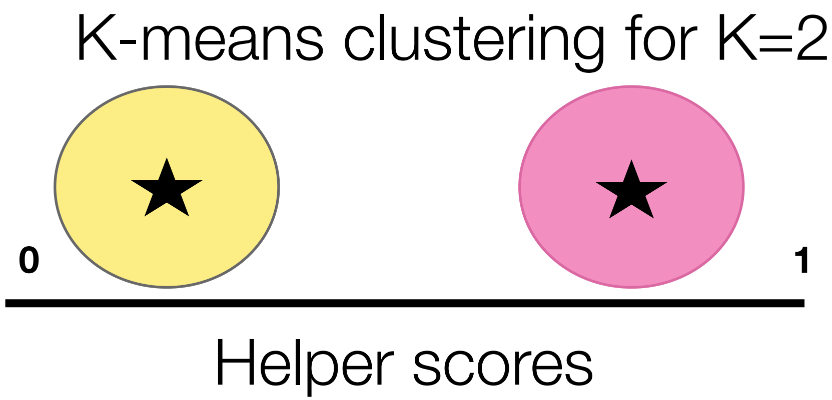
K-means clustering for K=1



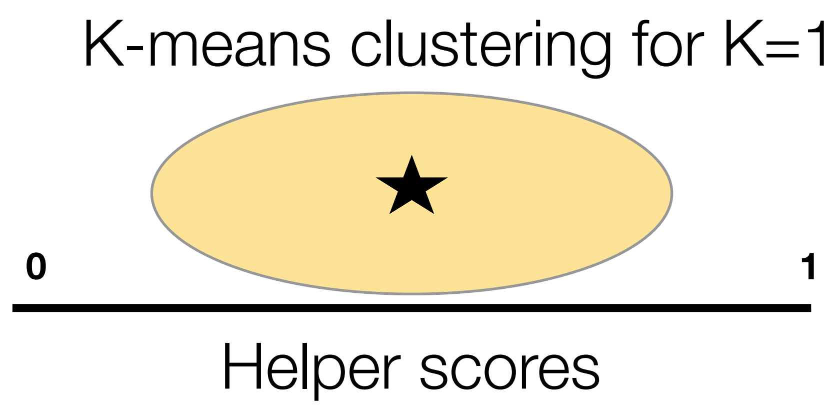
Helper scores

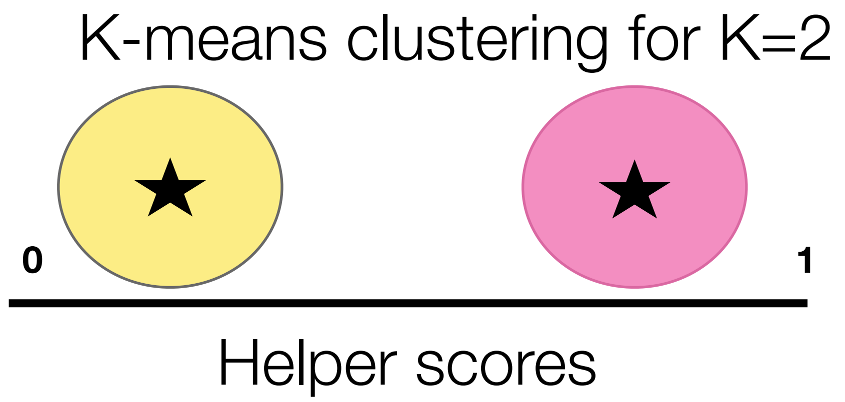
CHI: one or two clusters?





CHI: one or two clusters?

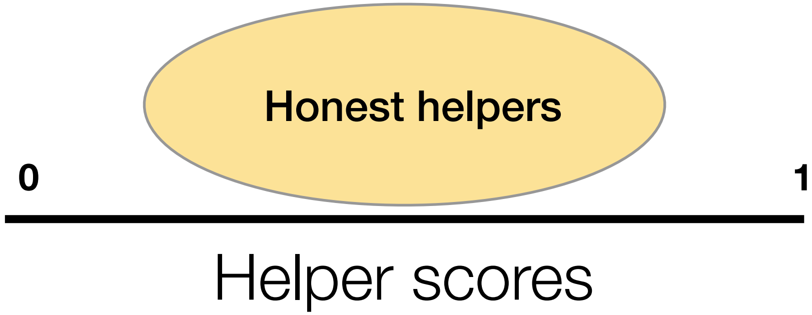




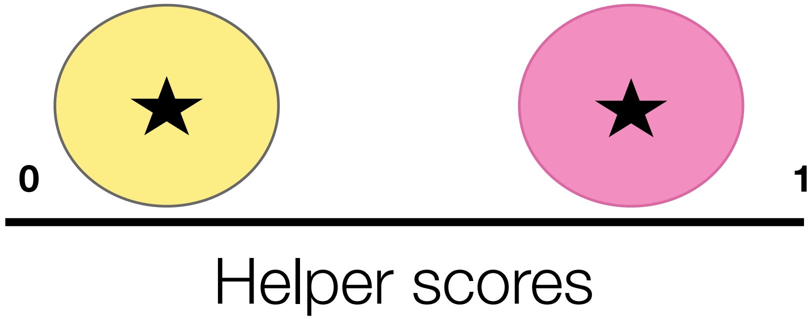
- Is the clustering accuracy **better above a threshold** for K=2 compared to K=1?

CHI: one or two clusters?

K-means clustering for K=1



K-means clustering for K=2

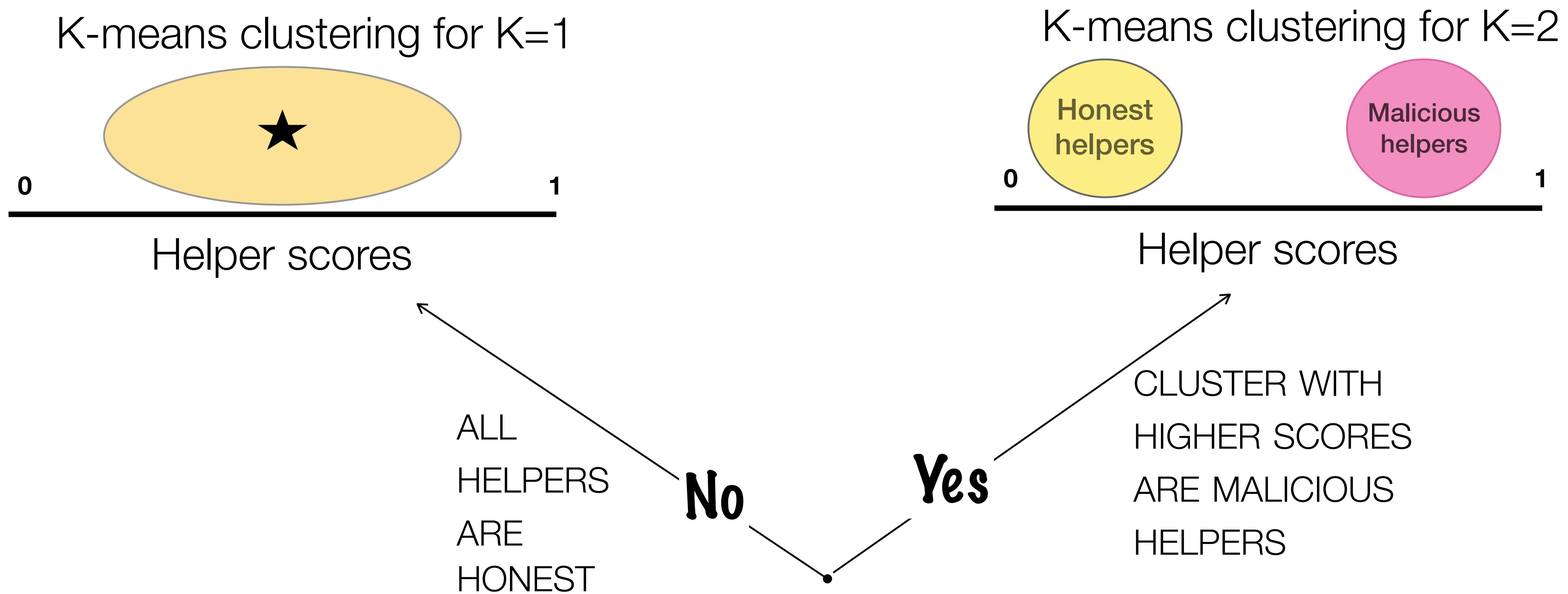


ALL
HELPERS
ARE
HONEST

No

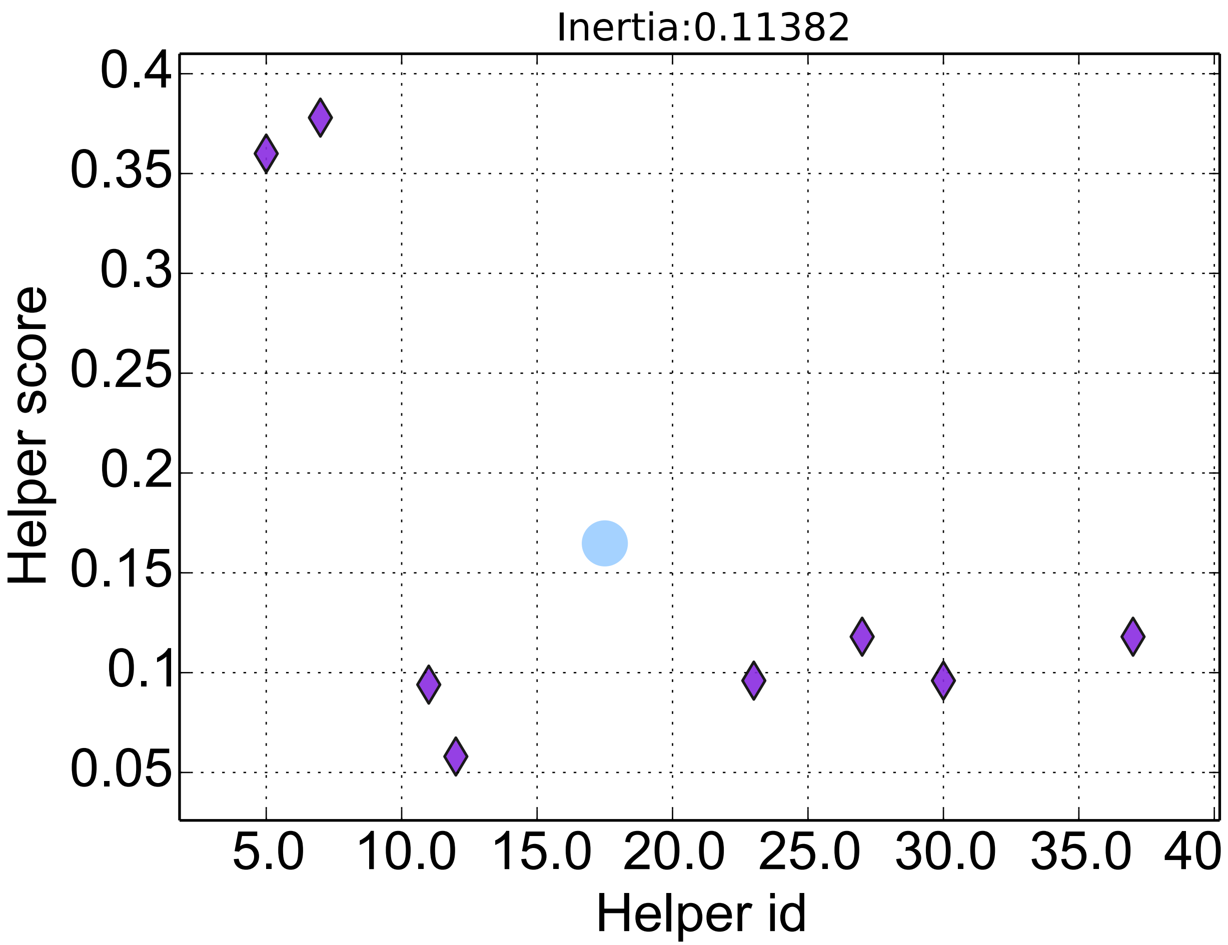
- Is the clustering accuracy **better above a threshold** for *K*=2 compared to *K*=1?

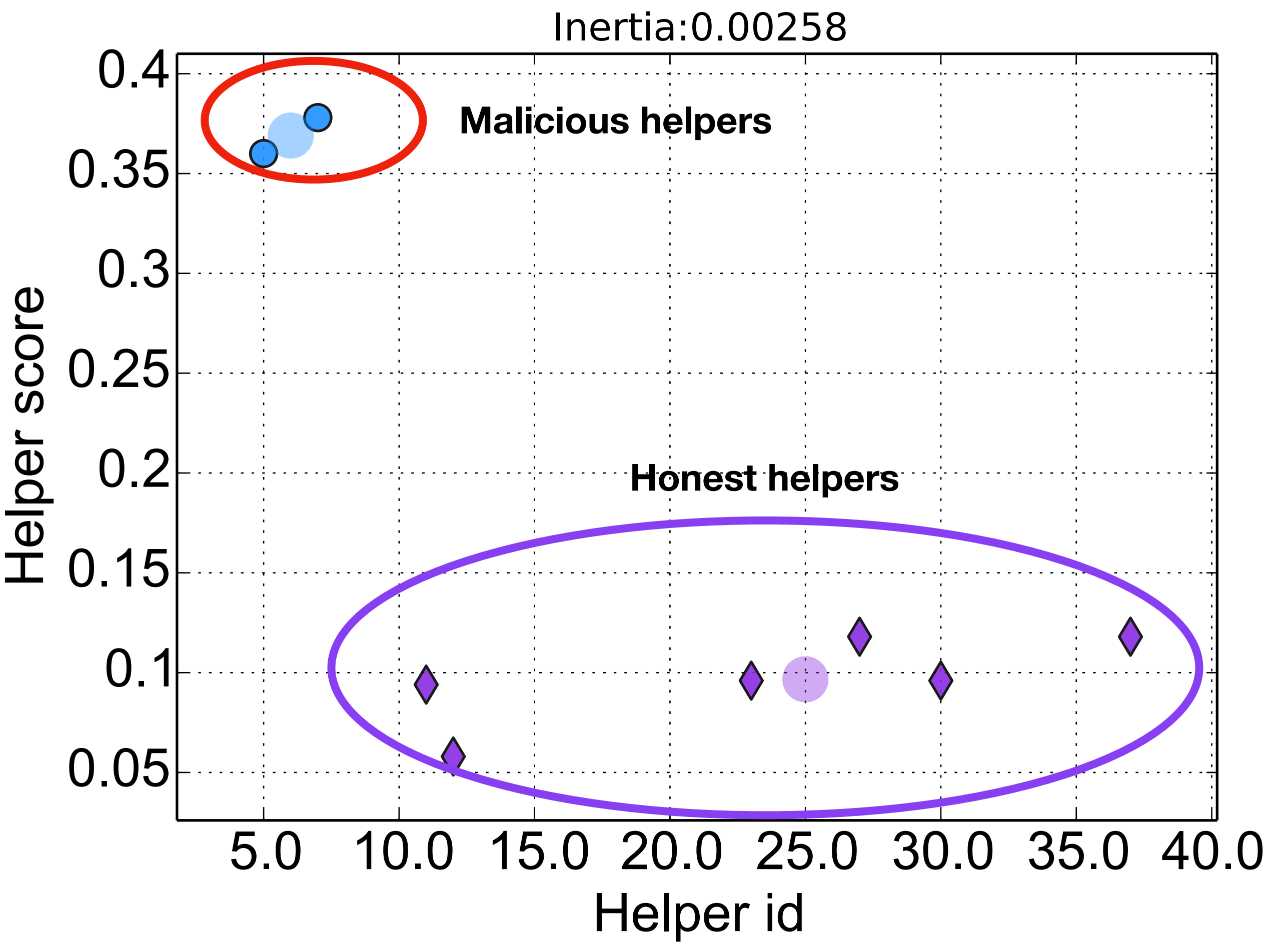
CHI: one or two clusters?



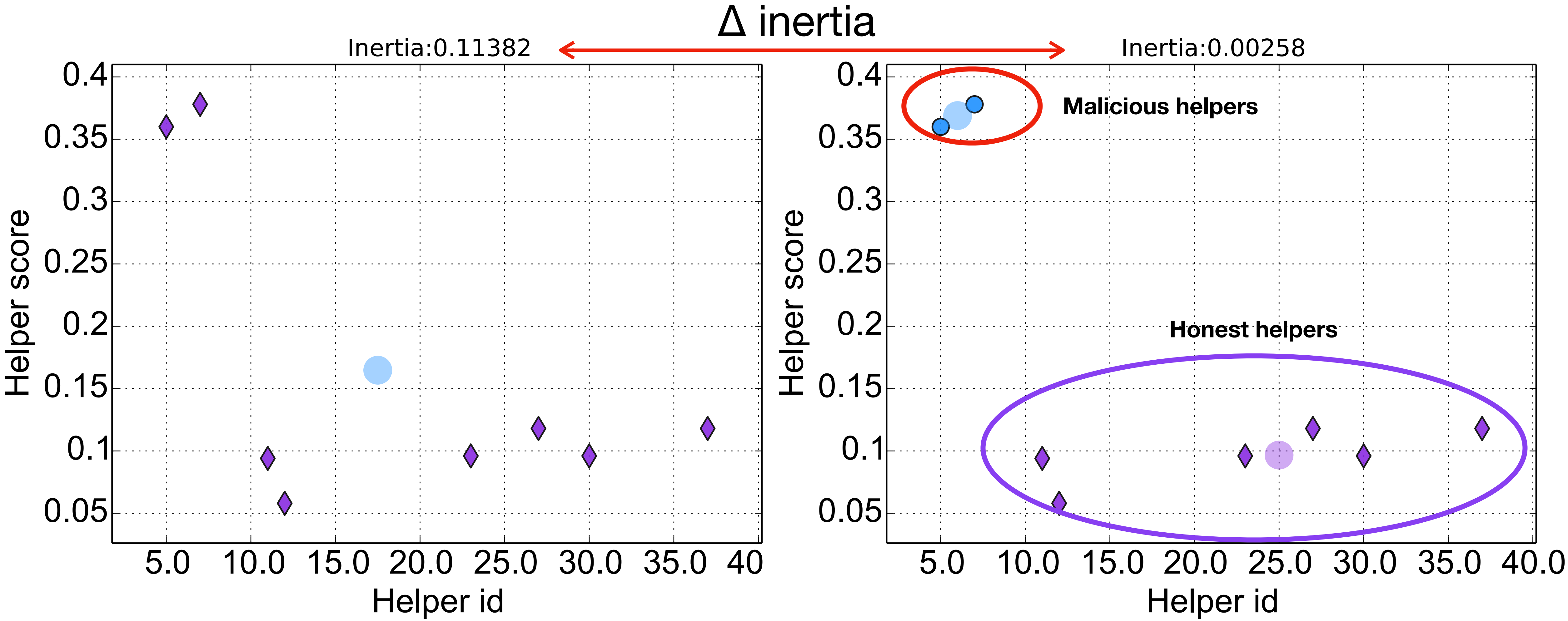
- Is the clustering accuracy **better above a threshold** for *K*=2 compared to *K*=1?

CHI clusters for K=1 and K=2

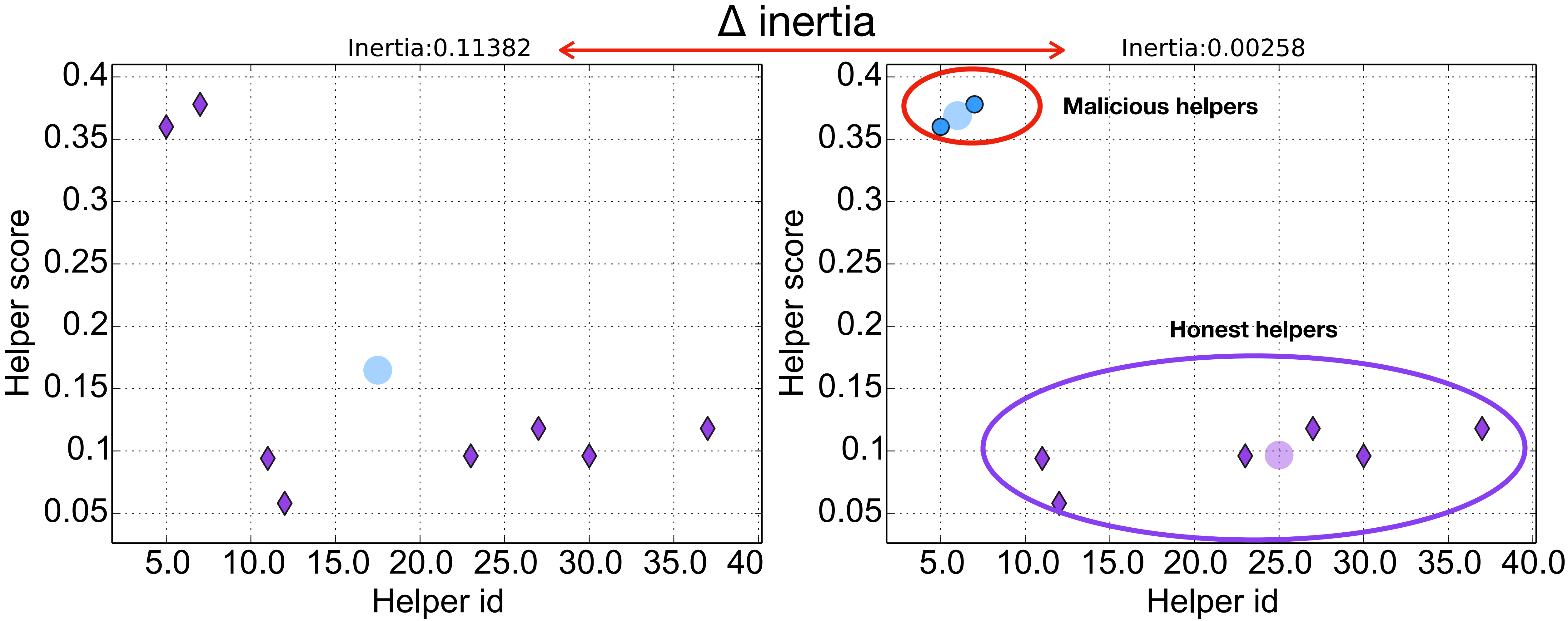




CHI clusters for K=1 and K=2

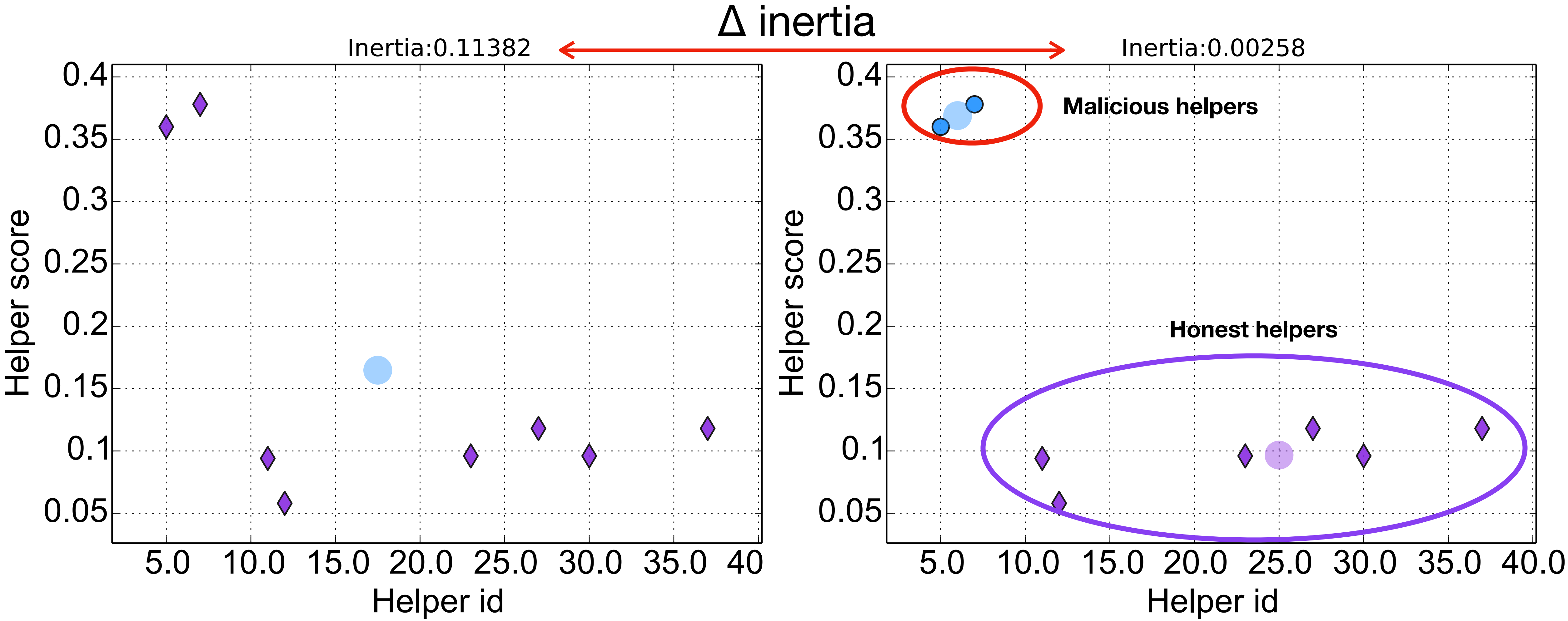


CHI clusters for K=1 and K=2



High inertia difference indicates the existence of malicious helper(s)

CHI clusters for K=1 and K=2



High inertia difference indicates the existence of malicious helper(s)

Spass

- Design goals
- Contract functionality
- Optimal contract parameters
- Malicious helper identification
- **Performance**
- Business feasibility of Spass

Spass

- Optimal number of helpers
- Robustness of CHI to malicious helpers
- Impact of lossy compression
- Sensing accuracy

- Design goals
- Contract functionality
- Optimal contract parameters
- Malicious helper identification
- **Performance**
- Business feasibility of Spass

Optimal number of helpers

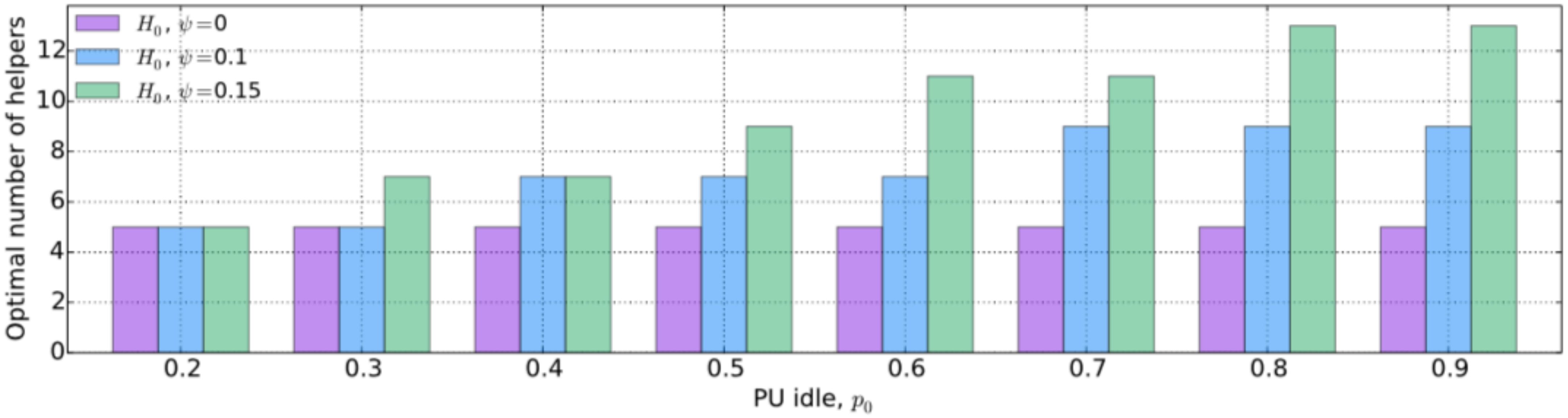
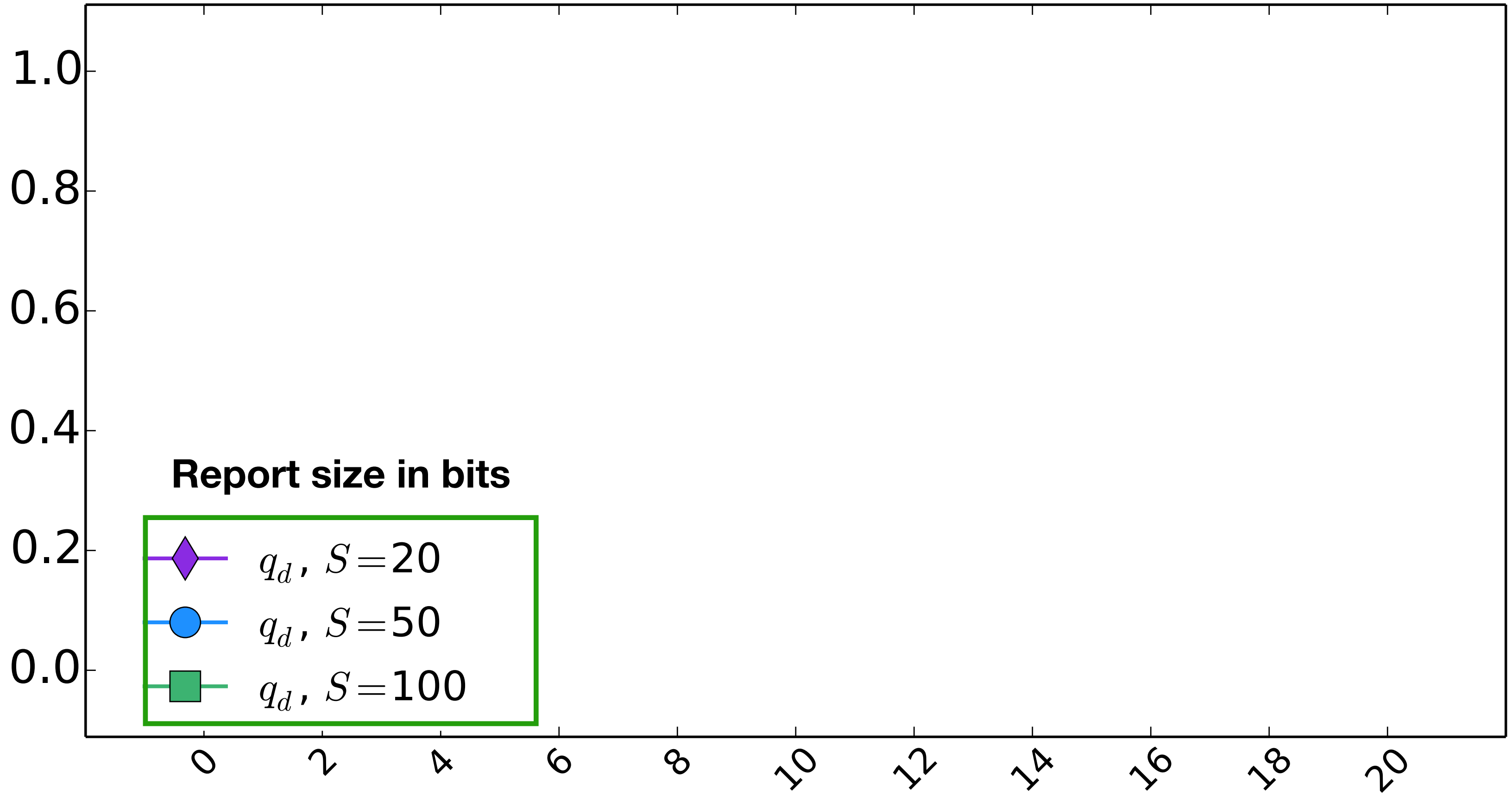


Fig. 8. Impact of increasing p_0 on optimal number of helpers for various ψ values. Following parameters are used: $R_s = 5, \mu_{eth} = 0.1, \mu_s = 0.05, \mu = 1$, and $\beta = 10, \kappa = 10$.

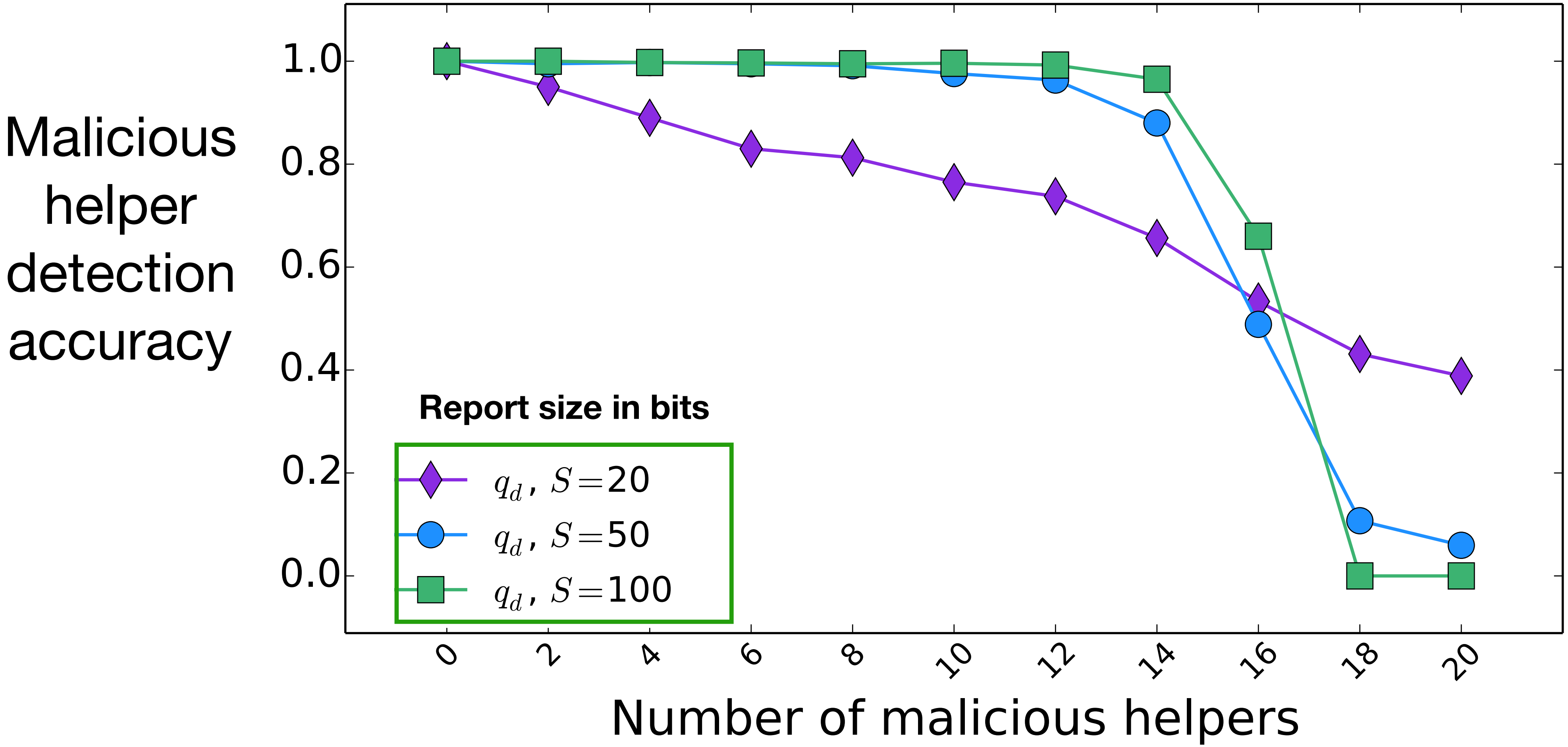
- Higher P_0, higher number of helpers
- More malicious helpers expected, more helpers need to be selected

Malicious helper detection accuracy



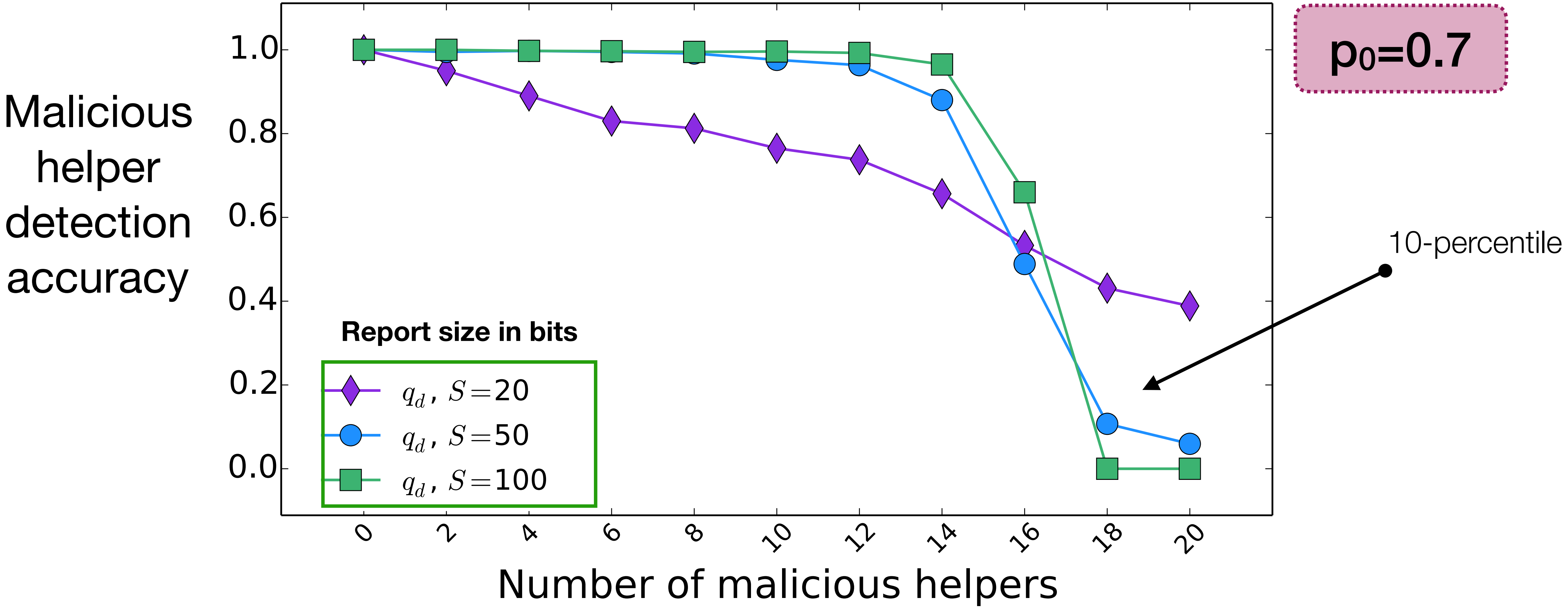
p_0=0.7

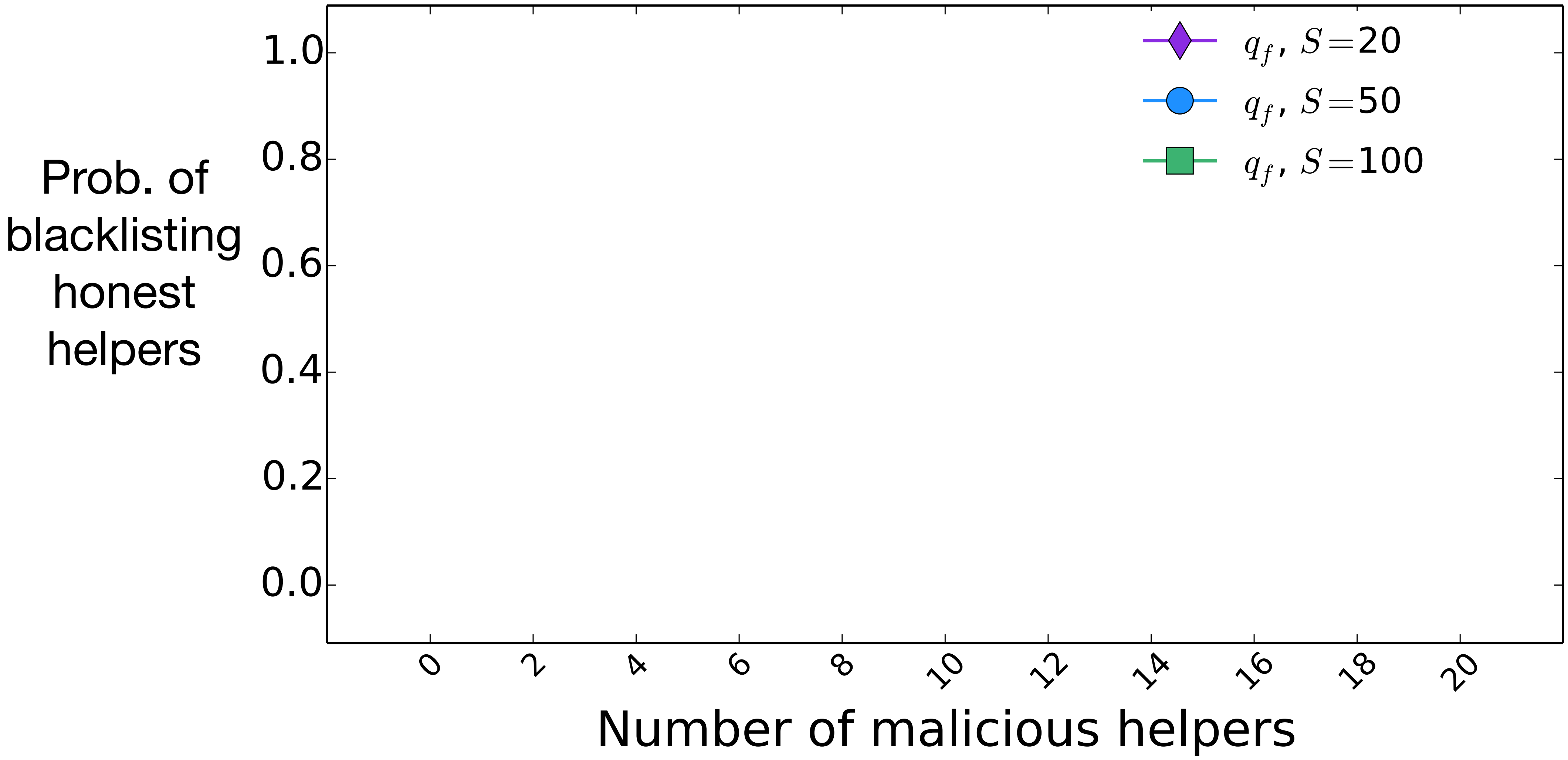
CHI achieves high malicious helper detection accuracy



p_0=0.7

CHI achieves high malicious helper detection accuracy

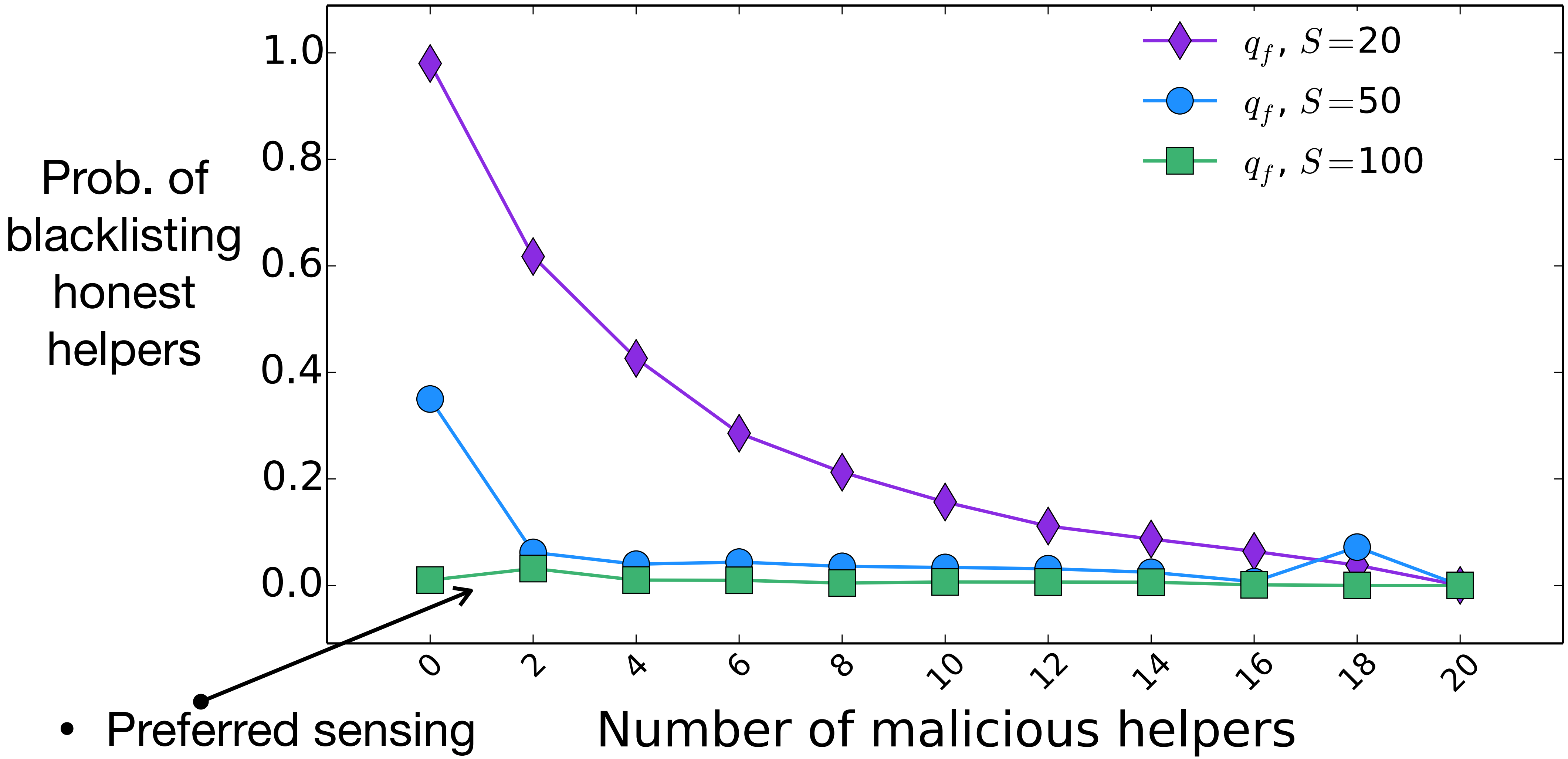




Prob. of blacklisting honest helpers
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
16
18
20
Number of malicious helpers
q_f, S=20
q_f, S=50
q_f, S=100

p_0=0.7

Almost always no false alarms if report size is bigger than 100 bits

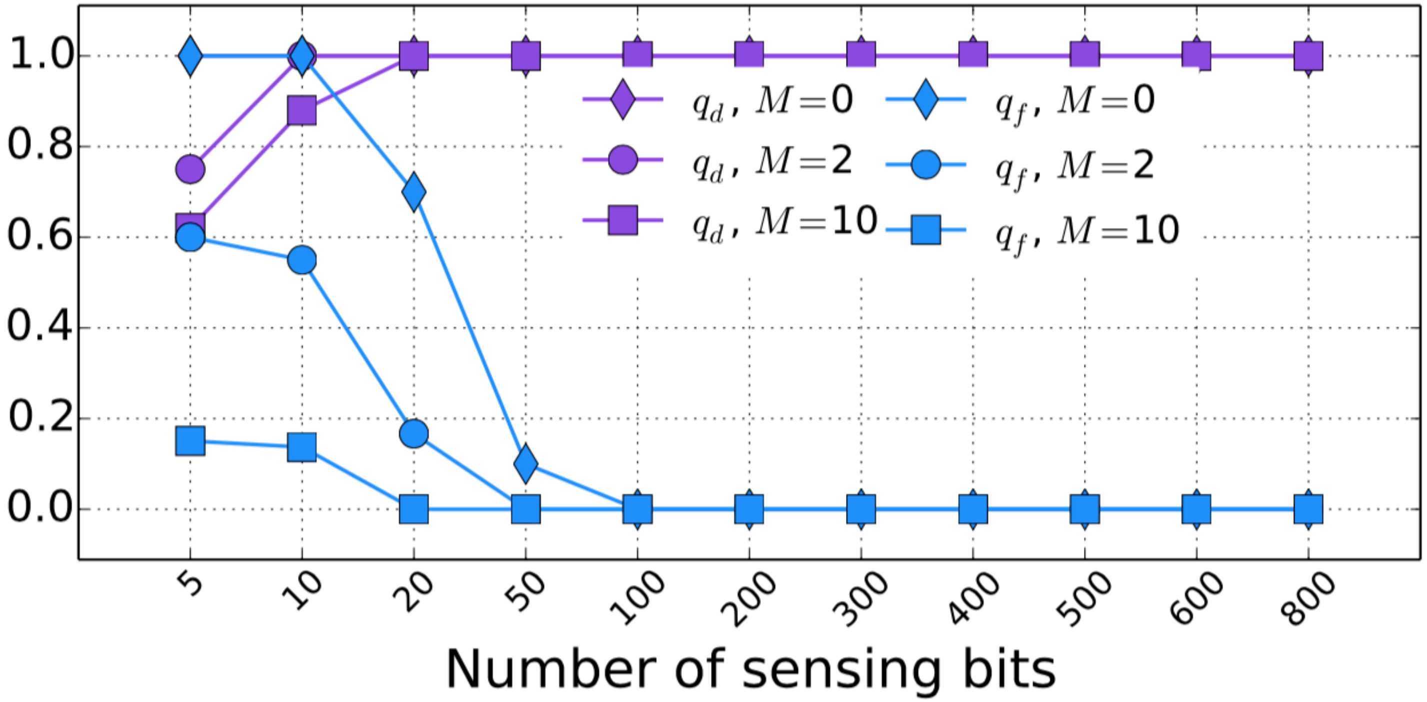


p_0=0.7

- Preferred sensing report size

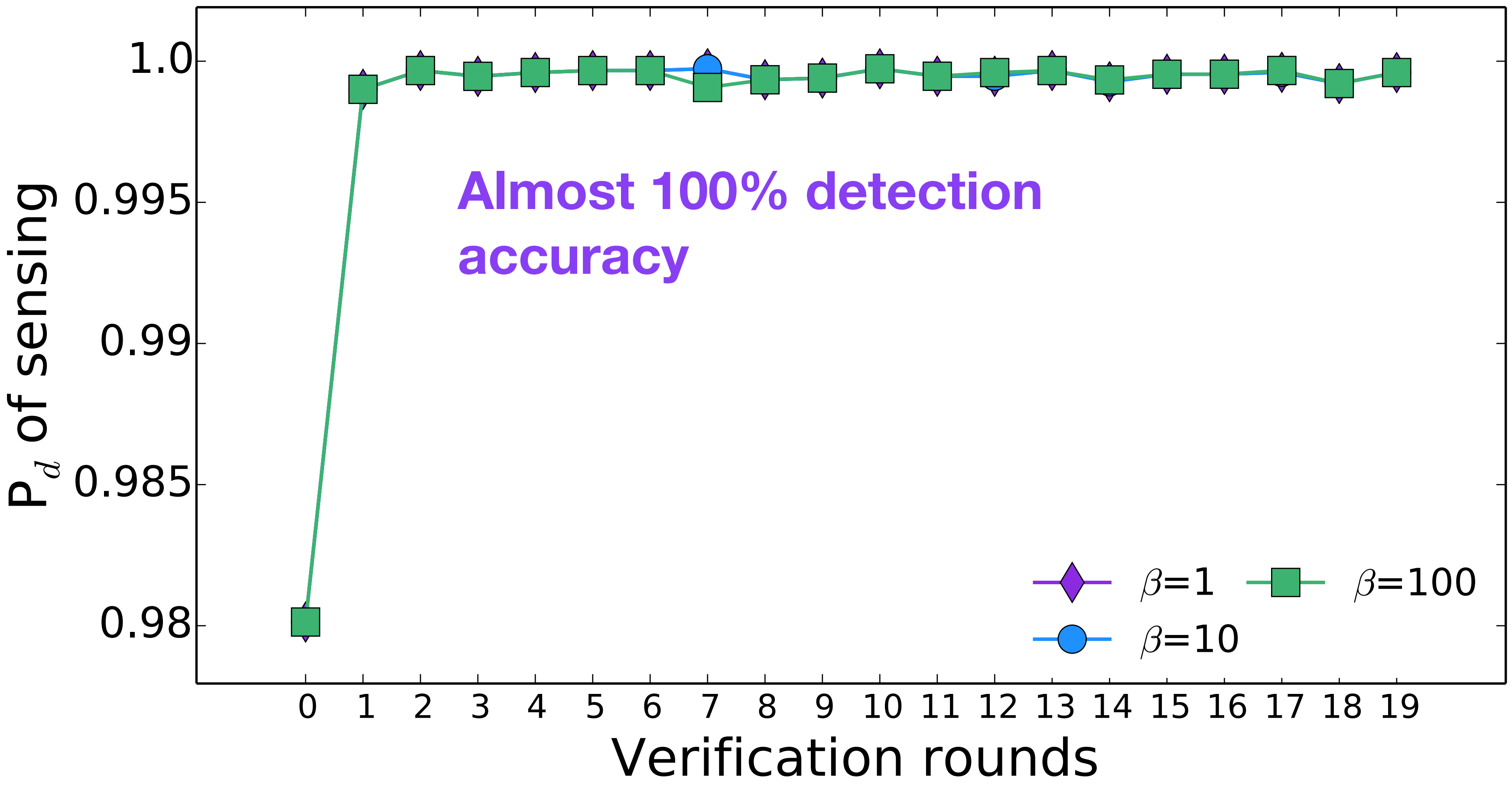
Minimum report size needed

CHI accuracy



(b) Impact of S on accuracy.

Spectrum discovery with very high accuracy



In a nutshell

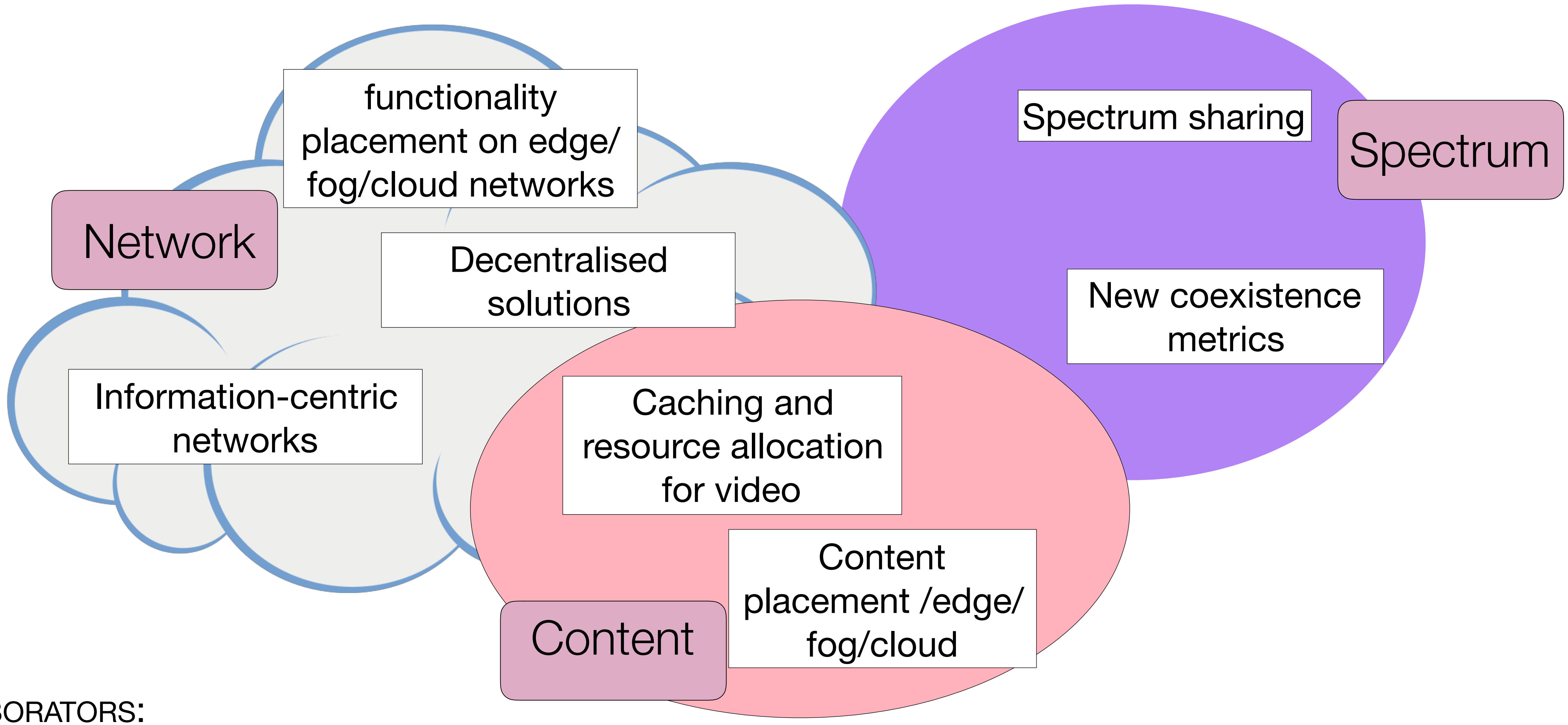
- Spass: https://github.com/zubow/Spass_contract
    - Spectrum sensing as a service using smart contracts (Ethereum)
    - Cost of smart contract usage and helpers sensing service
    - Spass becomes profitable under *some* conditions
    - Optimal number of helpers and verification round durations
- Possible directions:
    - More sophisticated malicious users
    - Pricing and reward assignment based on helper capabilities
    - Computation cost should also be considered

Crowdsourcing for spectrum sensing

- Spectrum monitoring for policy making and misuse detection
  - SpecGuard
- Radio Environment Map construction (spectrum sensing provider)
  - SpecSense
- Pricing for crowdsensing
  - Han et al.
- Malicious sensor identification
  - Catch Me If You Can
- Privacy aspects
  - DPSense

- Jin, Xiaocong, et al. "Specguard: Spectrum misuse detection in dynamic spectrum access systems." *IEEE TMC* 2018
- Chakraborty, Ayon, et al. "Specsense: Crowdsensing for efficient querying of spectrum occupancy." *IEEE INFOCOM 2017*
- DPSense: Differentially Private Crowdsourced Spectrum Sensing, ACM CCS 2018
- Ying, Xuhang, Sumit Roy, and Radha Poovendran. "Pricing mechanisms for crowd-sensed spatial-statistics-based radio mapping." *IEEE Transactions on Cognitive Communications and Networking, 2017*.
- Han, Kai, He Huang, and Jun Luo. "Quality-Aware Pricing for Mobile Crowdsensing." *IEEE/ACM TON* 2018
- Li, Husheng, and Zhu Han. "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks." *IEEE Transactions on Wireless Communications* 2010.

The future is **unlicensed**, **diverse**, and **decentralized**



COLLABORATORS:

TECHNISCHE UNIVERSITÄT MÜNCHEN

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
TIETOJENKÄSITTELYTIETEEN LAITOS
INSTITUTIONEN FÖR DATAVETENSKAP
DEPARTMENT OF COMPUTER SCIENCE

YARMOUK UNIVERSITY

Sabancı Üniversitesi

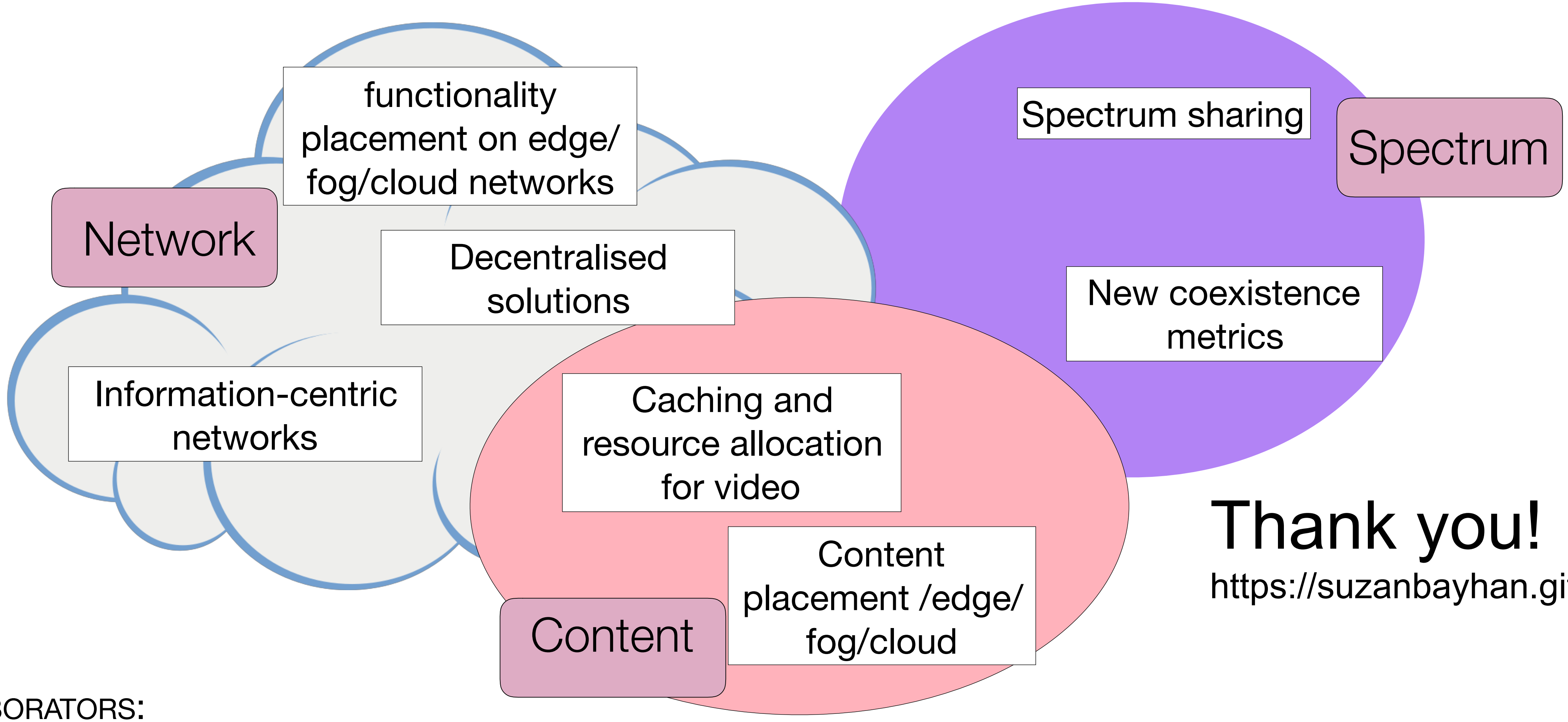
Zurich University of Applied Sciences
zh aw School of Engineering

Technische Universität Berlin

BOĞAZİÇİ ÜNİVERSİTESİ 1863

UNIVERSITY OF CAMBRIDGE

The future is **unlicensed**, **diverse**, and **decentralized**



Thank you!

https://suzanbayhan.github.io/

COLLABORATORS:

Technische Universität München
Helsingin yliopisto Helsingfors universitet University of Helsinki
Tietojenkäsittelytieteen laitos Institutionen för datavetenskap Department of Computer Science
Yarmouk University
Sabancı Universitesi
Zurich University of Applied Sciences zh aw School of Engineering
Technische Universität Berlin
Boğaziçi Üniversitesi 1863
University of Cambridge

Talk Abstract

To cope with the huge increase in cellular data traffic, mobile network operators (MNO) consider using also other spectrum bands which are under-utilized spatiotemporally. To discover such spectrum opportunities, several studies suggest exploiting the ubiquity of mobile devices to perform spectrum sensing. However, current solutions for crowdsourced spectrum sensing overlook the fact that mobile devices lack the incentives to sense without certain benefits as spectrum sensing results in energy and CPU overhead. To encourage participation, we explore the use of smart contracts running on a distributed ledger (e.g., Ethereum) for an MNO to announce its need for spectrum sensing, the interested sensors to register themselves as candidate sensors, and MNO to pay the selected sensors for their service. Despite the simplicity of the idea, the realization requires more thoughts as smart-contracts are limited in their storage and processing capacity, and incurs a high cost for the write operation. I will introduce our proposal Spass: spectrum sensing as a service via smart contracts and address the following questions: (i) what is the cost of running smart contract based spectrum discovery? and (ii) given that both the helpers and the miners have to be paid, under which conditions an MNO can sustain a profitable business via smart-contract based spectrum discovery? (iii) How can the smart-contract catch free-riders among the sensing participants?

Bio:Suzan Bayhan is a senior researcher at TU Berlin and a docent in Computer Science at the University of Helsinki. She got her Ph.D. from Bogazici University in 2012 and worked as a post-doc at the University of Helsinki between 2012-2016. She was on N2Women board between 2017-2018. Her current research interests are resource allocation for wireless networks, spectrum sharing, and edge/fog/cloud computing. Suzan will join the University of Twente on September 2019 as an Assistant Professor.