Shuffle Private Linear Contextual Bandits

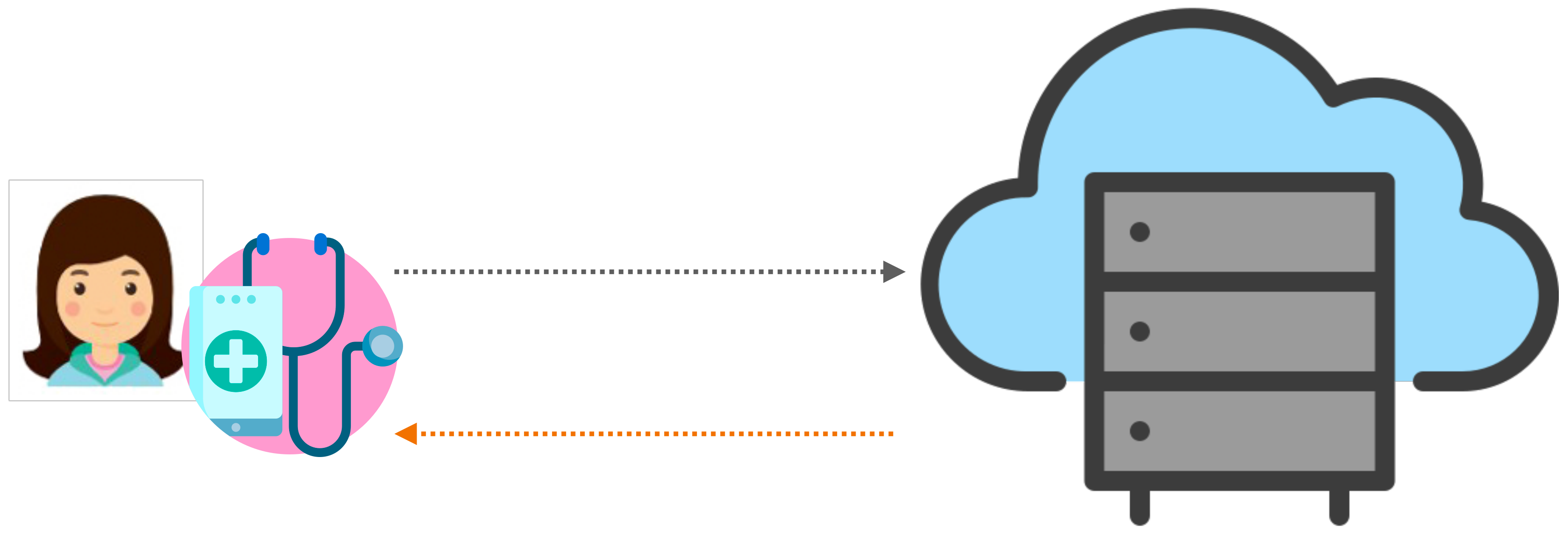
Xingyu Zhou, Sayak Ray Chowdhury*

Wayne State University

ICML'22

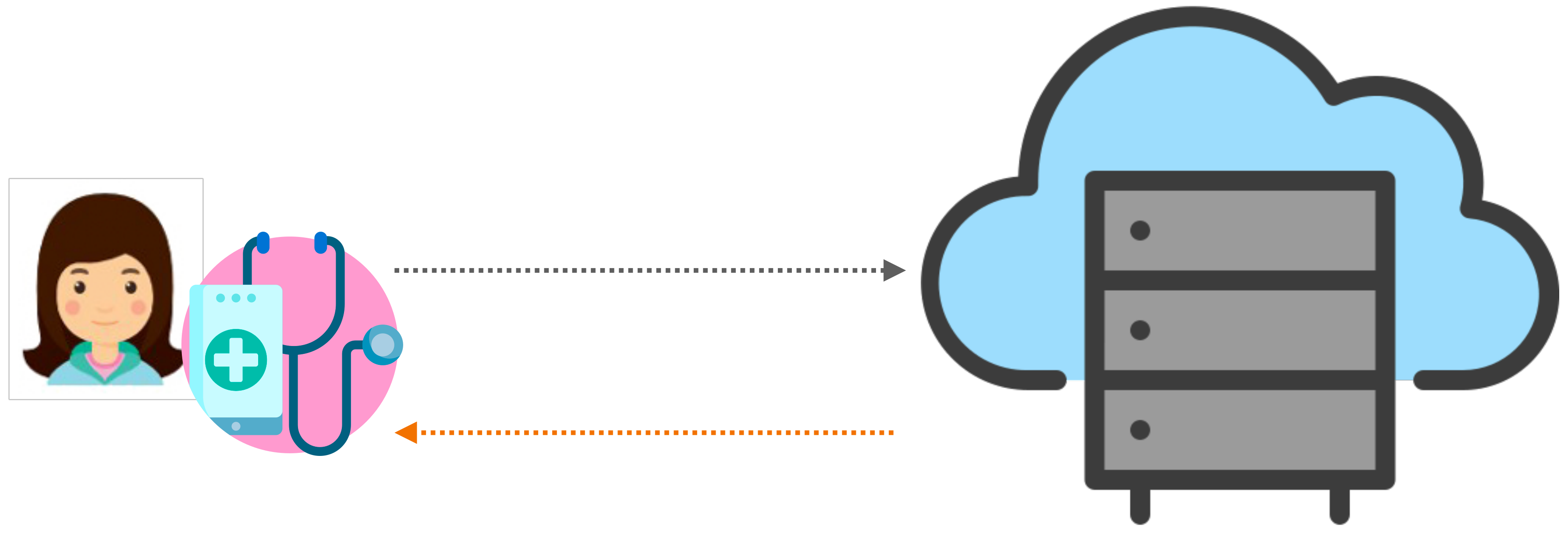
* Equal Contributions, Post-doc at Boston University

Linear Contextual Bandits (LCB)



Linear Contextual Bandits (LCB)

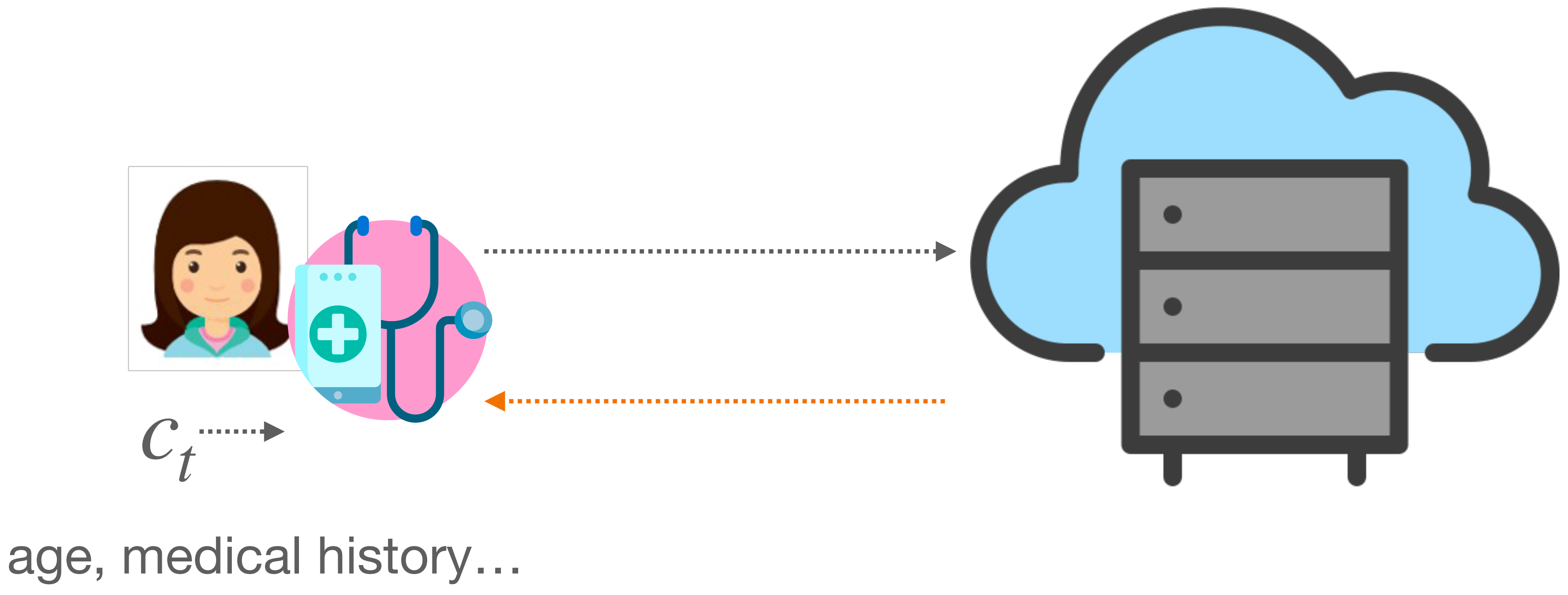
- For each time $t = 1, \ldots, T$



Linear Contextual Bandits (LCB)

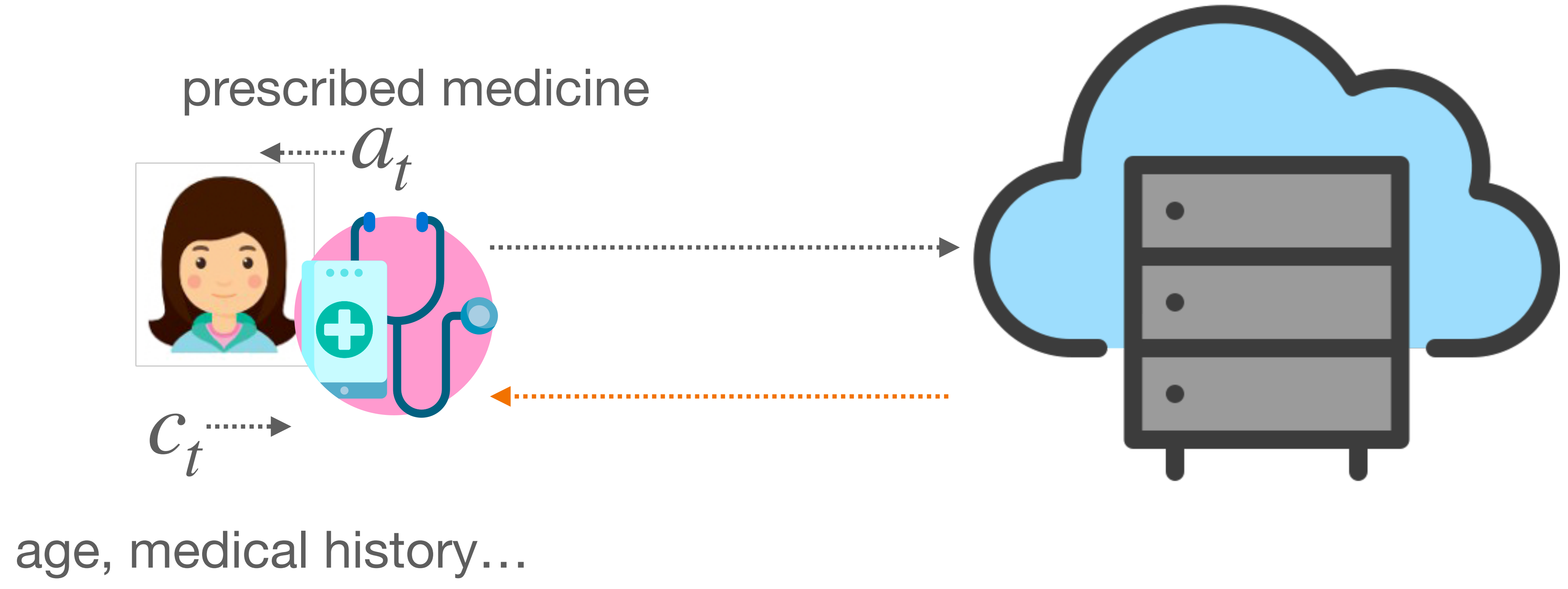
- For each time $t = 1, \ldots, T$

    1. Observe context c_t



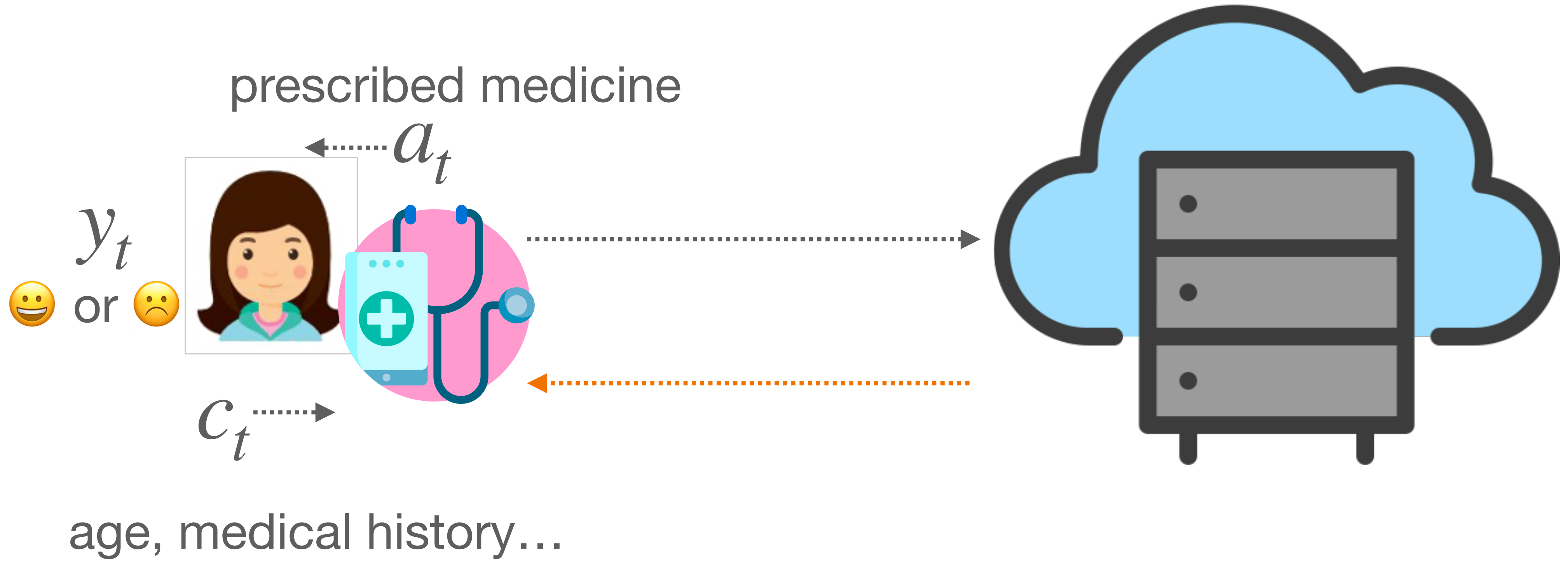
Linear Contextual Bandits (LCB)

- For each time $t = 1,\ldots,T$
  1. Observe context c_t
  2. Prescribes action a_t



Linear Contextual Bandits (LCB)

- For each time $t = 1, \ldots, T$
  1. Observe context c_t
  2. Prescribes action a_t
  3. Receive reward $y_t = \langle \phi(c_t, a_t), \theta^* \rangle + \epsilon_t$



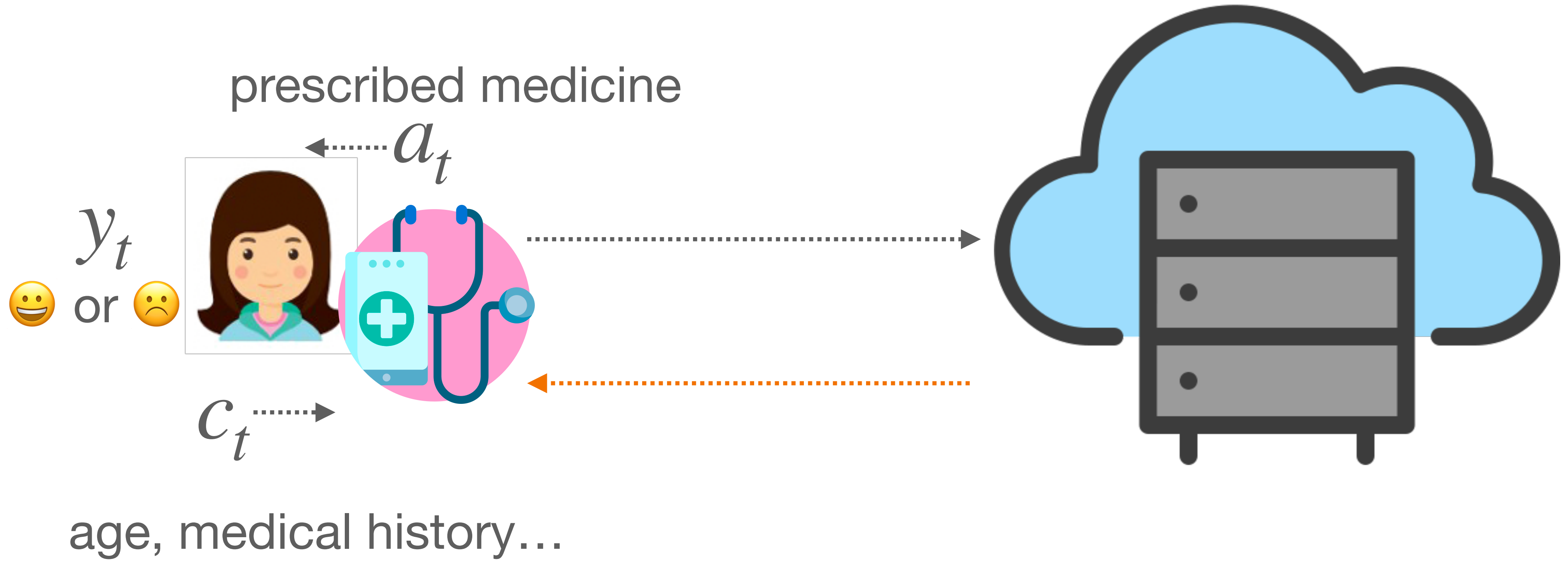
Linear Contextual Bandits (LCB)

- For each time $t = 1,\ldots,T$
  1. Observe context c_t
  2. Prescribes action a_t
  3. Receive reward $y_t = \langle \phi(c_t, a_t), \theta^* \rangle + \epsilon_t$

Known feature map

Noise

Unknown $\mathbb{R}^d$ vector



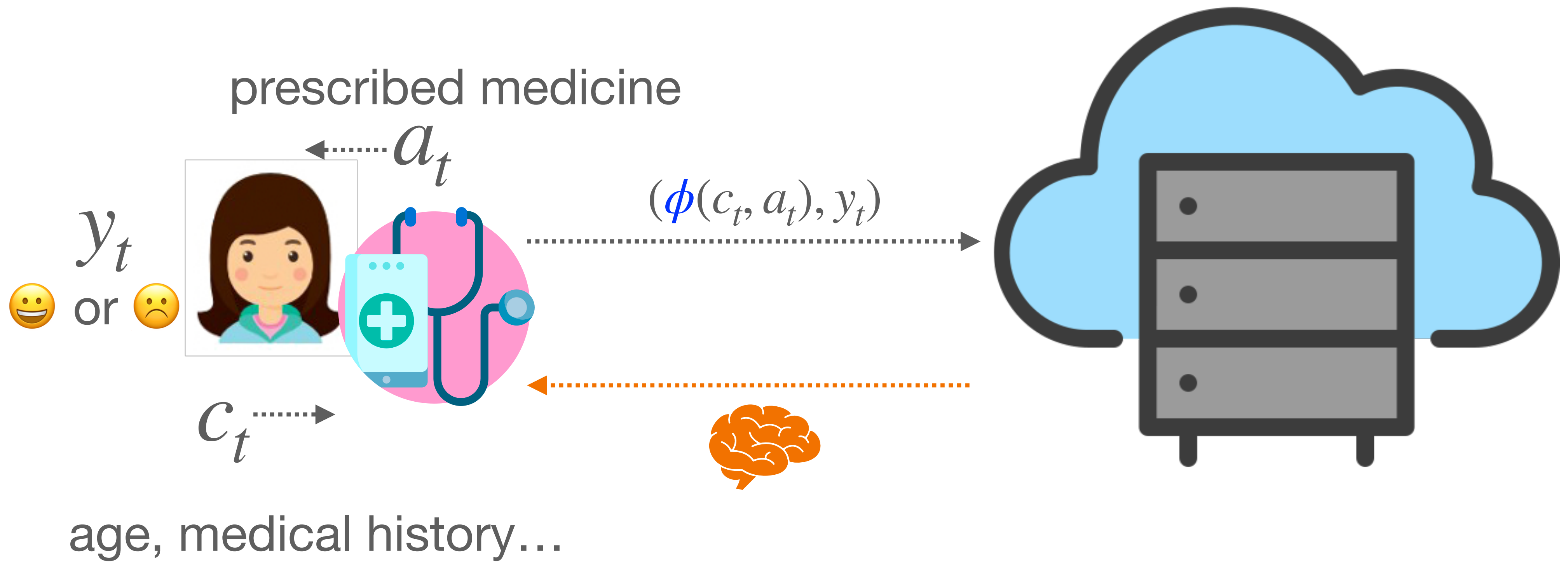
Linear Contextual Bandits (LCB)

- For each time $t = 1, \ldots, T$
  1. Observe context c_t
  2. Prescribes action a_t
  3. Receive reward $y_t = \langle \phi(c_t, a_t), \theta^* \rangle + \epsilon_t$
  4. Update model

Known feature map

Noise

Unknown $\mathbb{R}^d$ vector



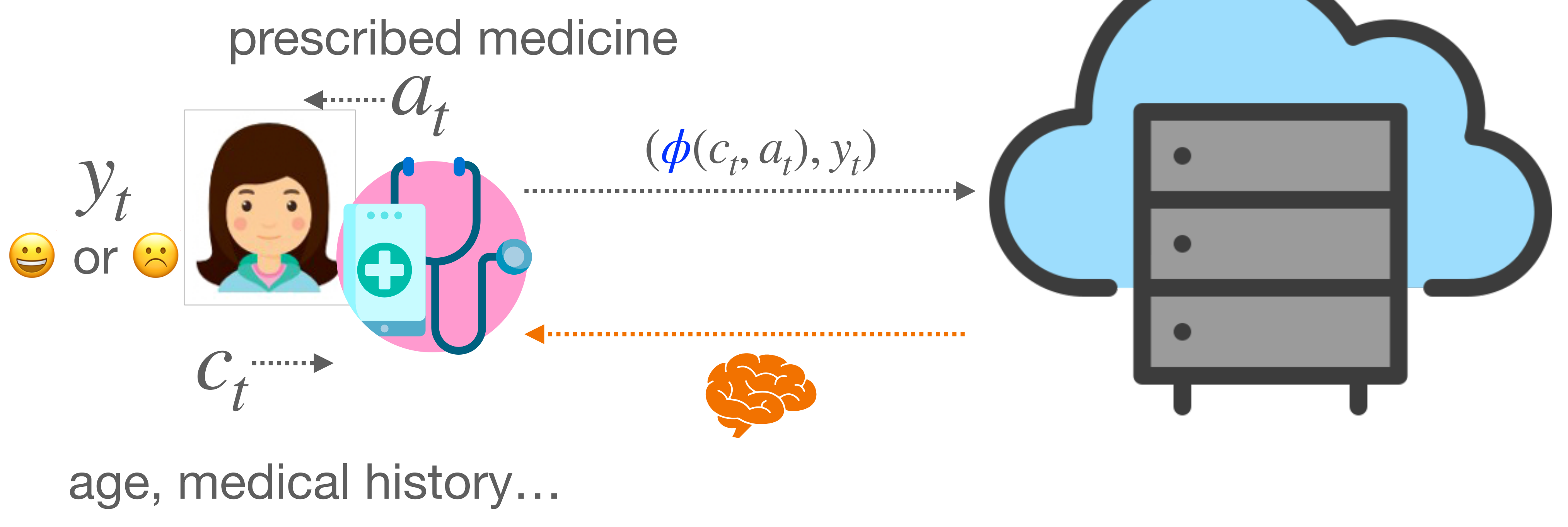
Linear Contextual Bandits (LCB)

- For each time $t = 1, \ldots, T$
  1. Observe context c_t
  2. Prescribes action a_t
  3. Receive reward $y_t = \langle \phi(c_t, a_t), \theta^* \rangle + \epsilon_t$
  4. Update model

Known feature map

Noise

Unknown $\mathbb{R}^d$ vector



- The goal is to minimize regret

$$\text{Reg}(T) = \sum_{t=1}^{T} \left[\max_a \langle \theta^*, \phi(c_t, a) \rangle - \langle \theta^*, \phi(c_t, a_t) \rangle \right]$$

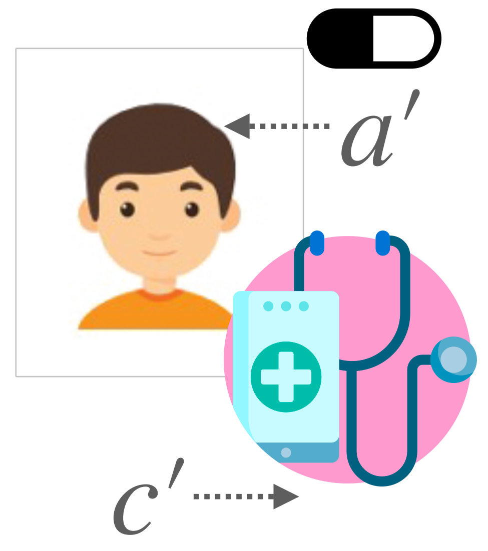
Privacy Risk

Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information

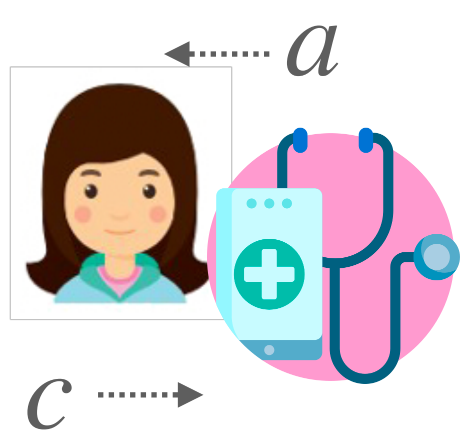
Privacy Risk

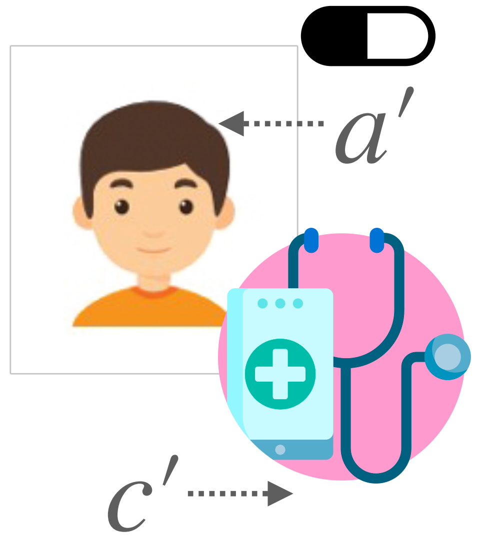
- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
    - Bob has diabetes and health app often prescribes



Privacy Risk

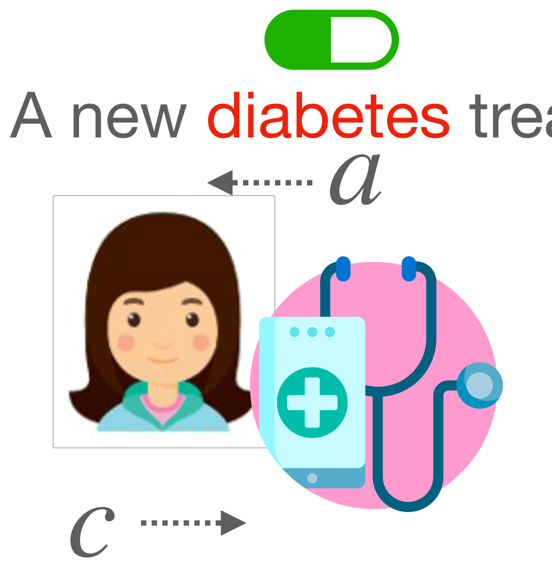
- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with

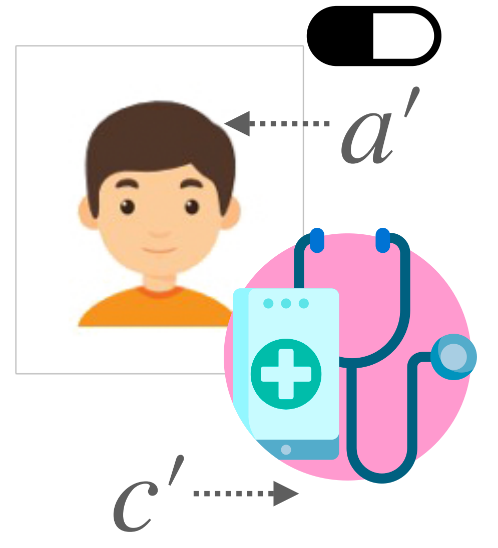




Privacy Risk

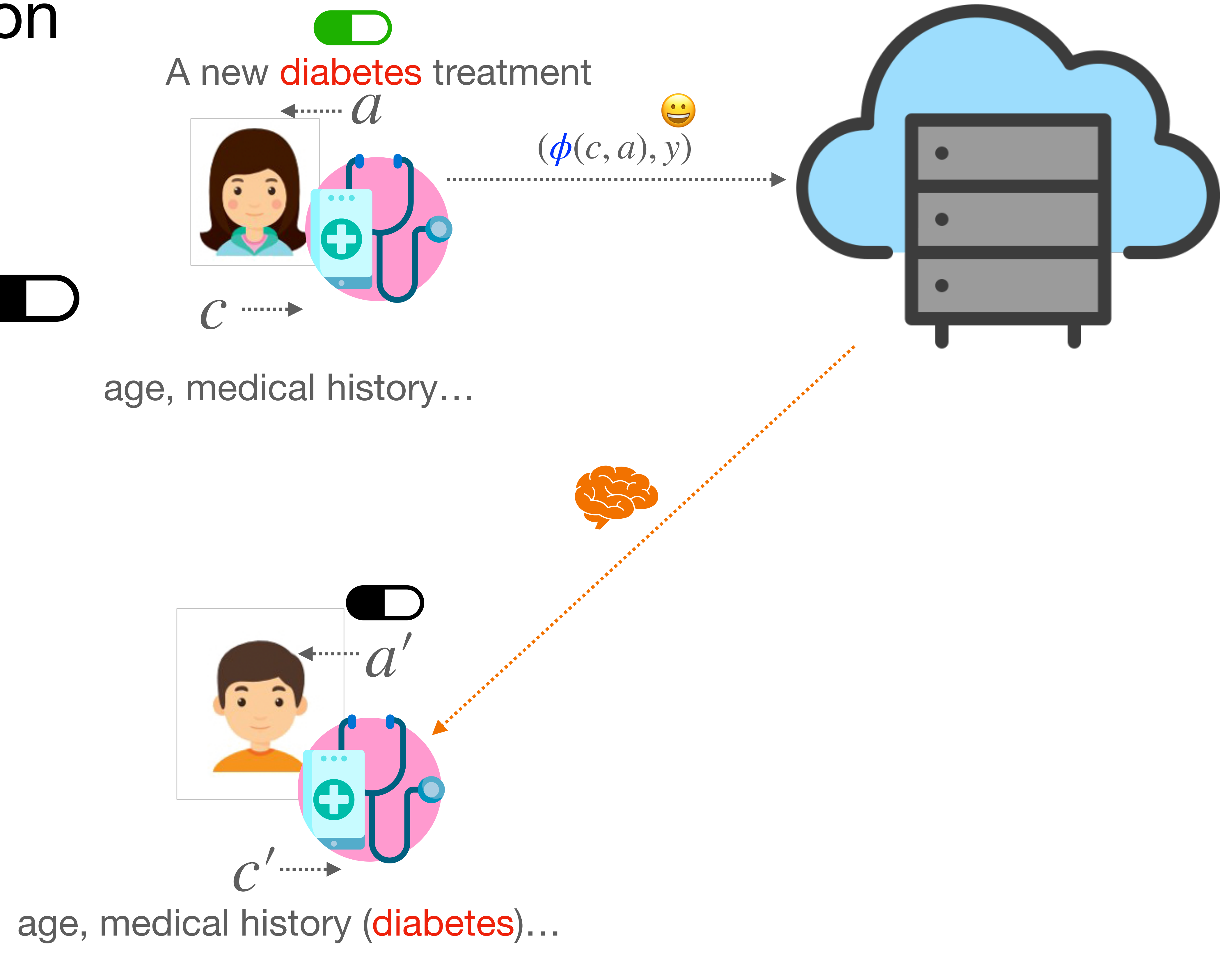
- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with





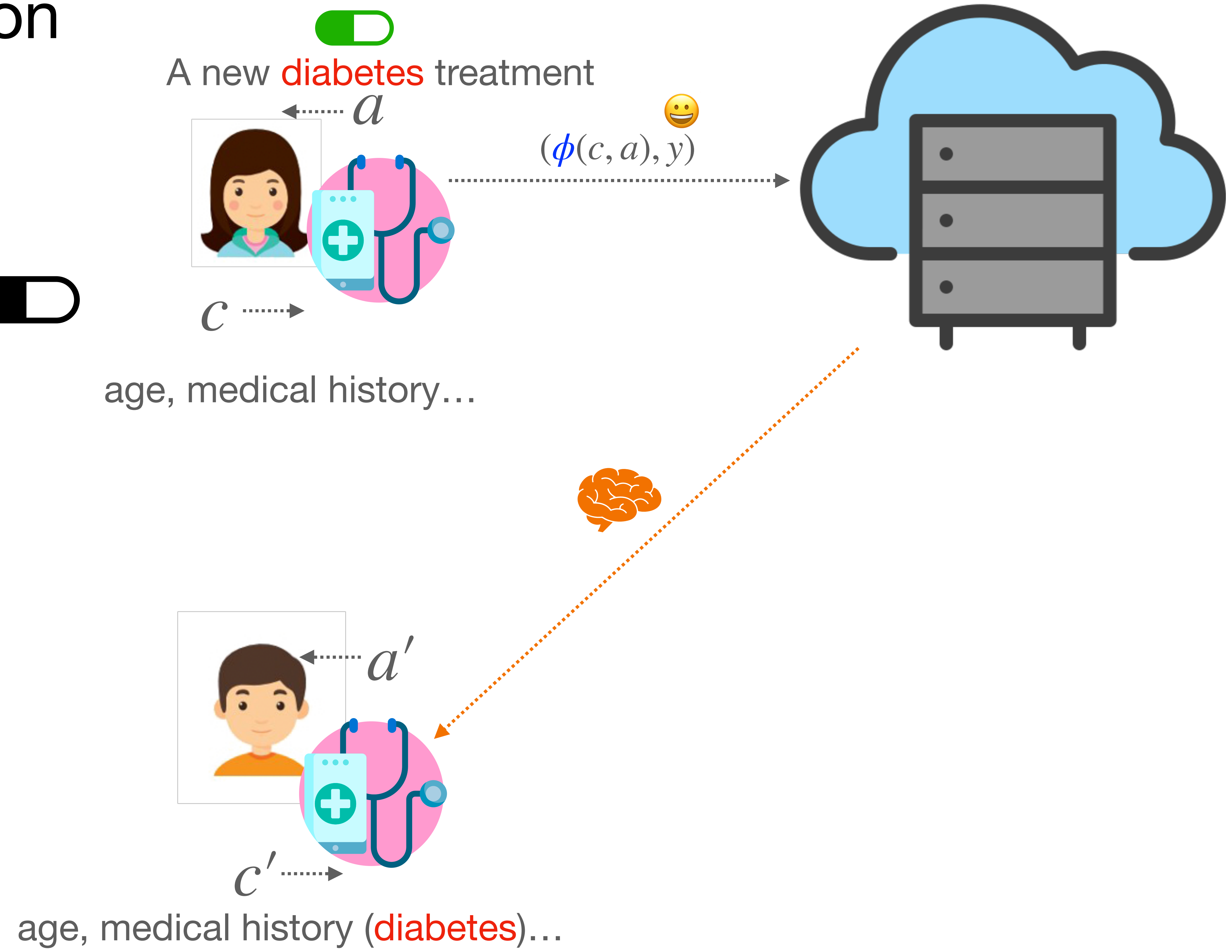
Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with



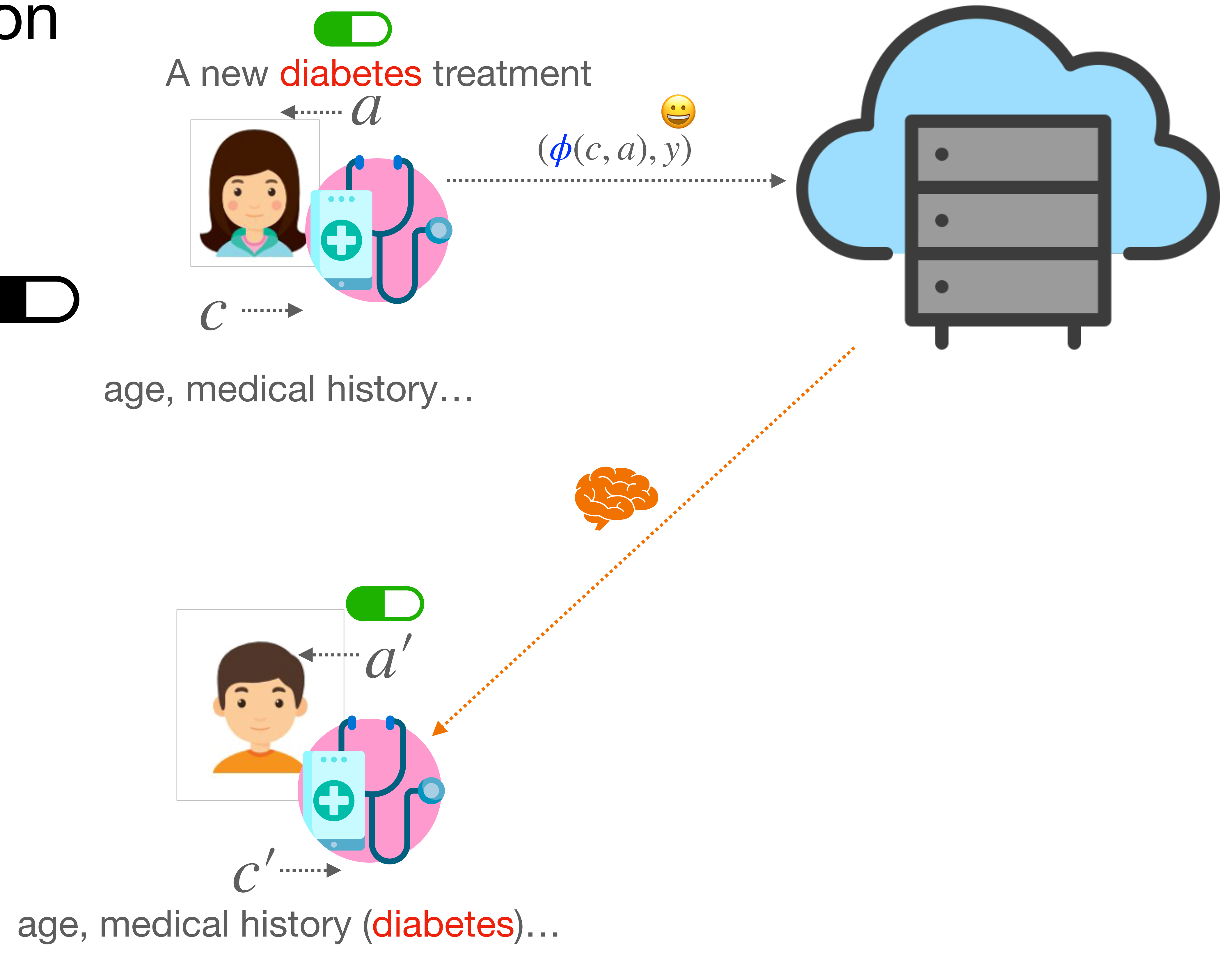
Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with
  - Bob receives new recommendation



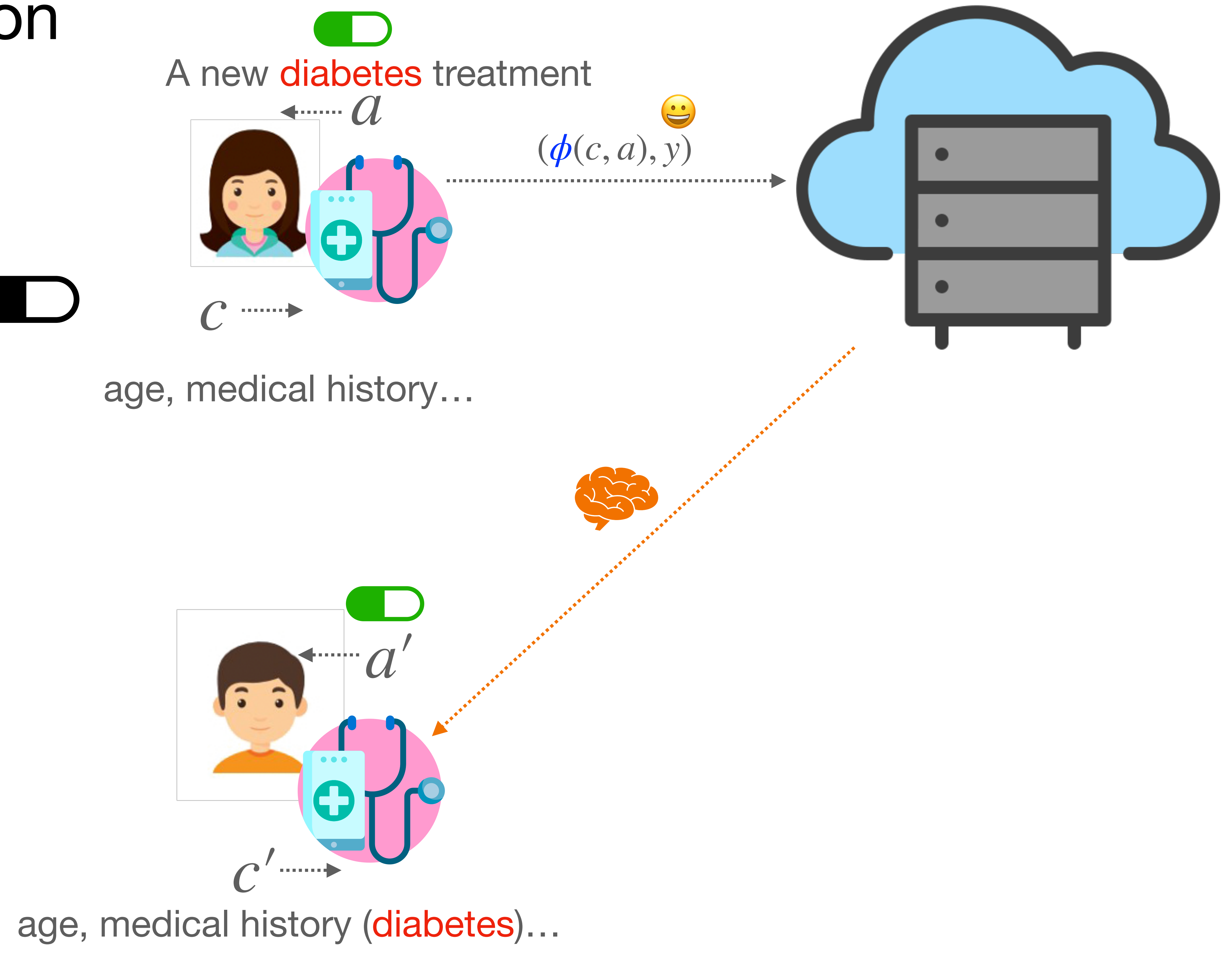
Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
    - Bob has diabetes and health app often prescribes
    - Alice is a **new** user and extremely happy with
    - Bob receives new recommendation



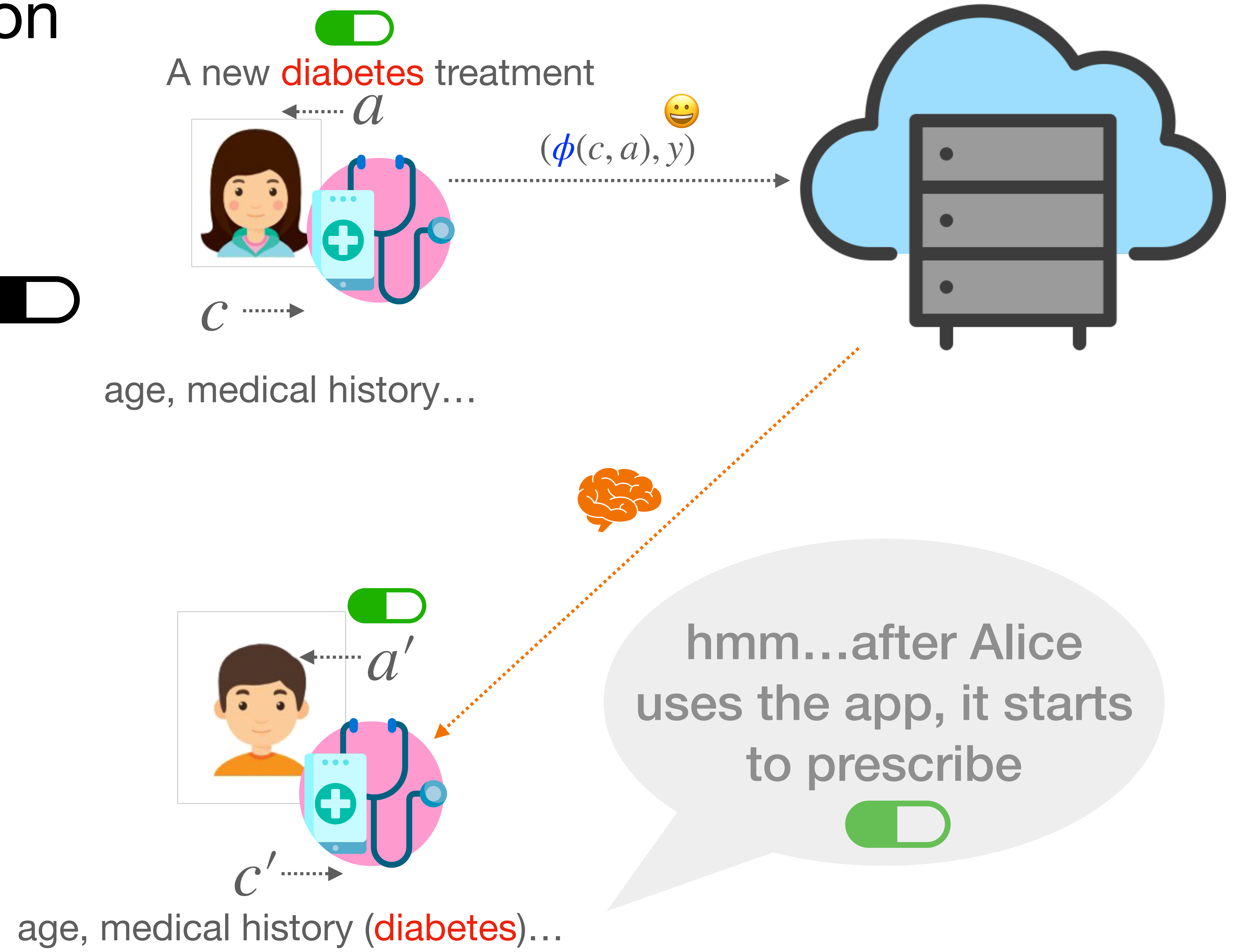
Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with
  - Bob receives new recommendation
  - *If Bob knows Alice is the most recent user*



Privacy Risk

- Both **context** and **reward** are sensitive information
- Standard LCB could reveal these information
  - Bob has diabetes and health app often prescribes
  - Alice is a **new** user and extremely happy with
  - Bob receives new recommendation
  - *If Bob knows Alice is the most recent user*
    - Bob's belief that Alice has diabetes **increases**



Differentially Private LCB

Central model

Differentially Private LCB

Central model

- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]

Differentially Private LCB

Central model

- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution

Differentially Private LCB

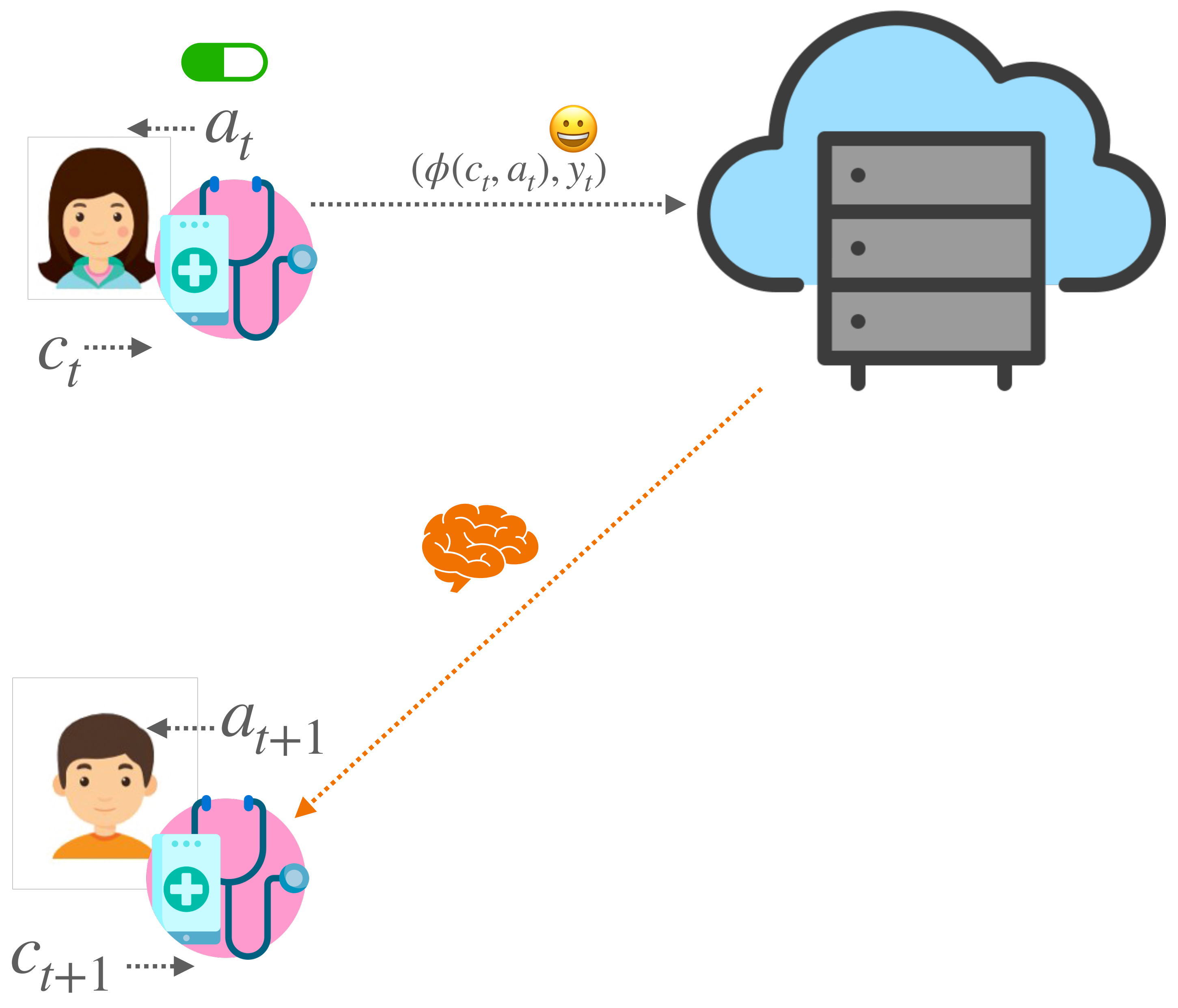
Central model

- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise

Differentially Private LCB

Central model

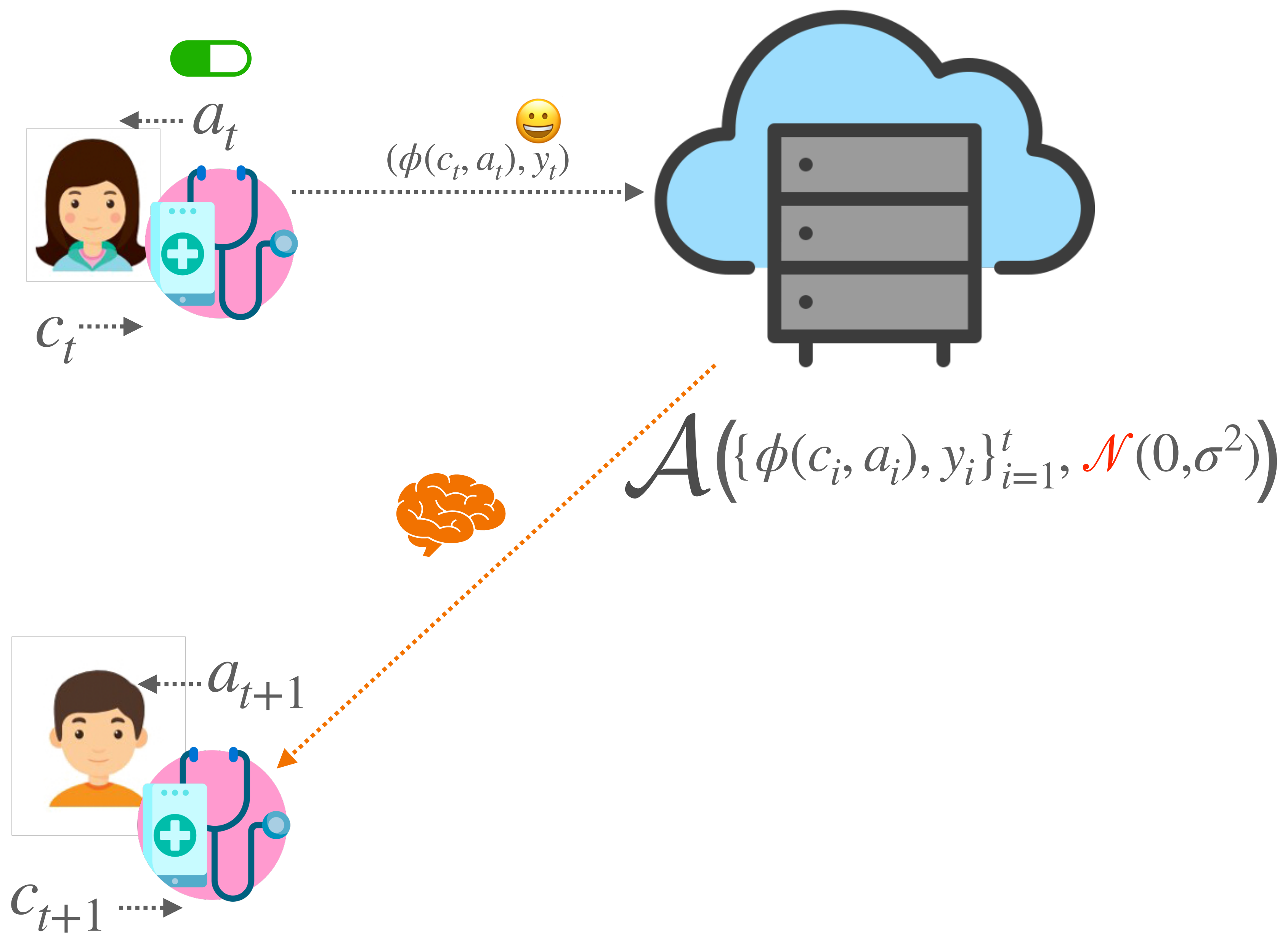
- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise



Differentially Private LCB

Central model

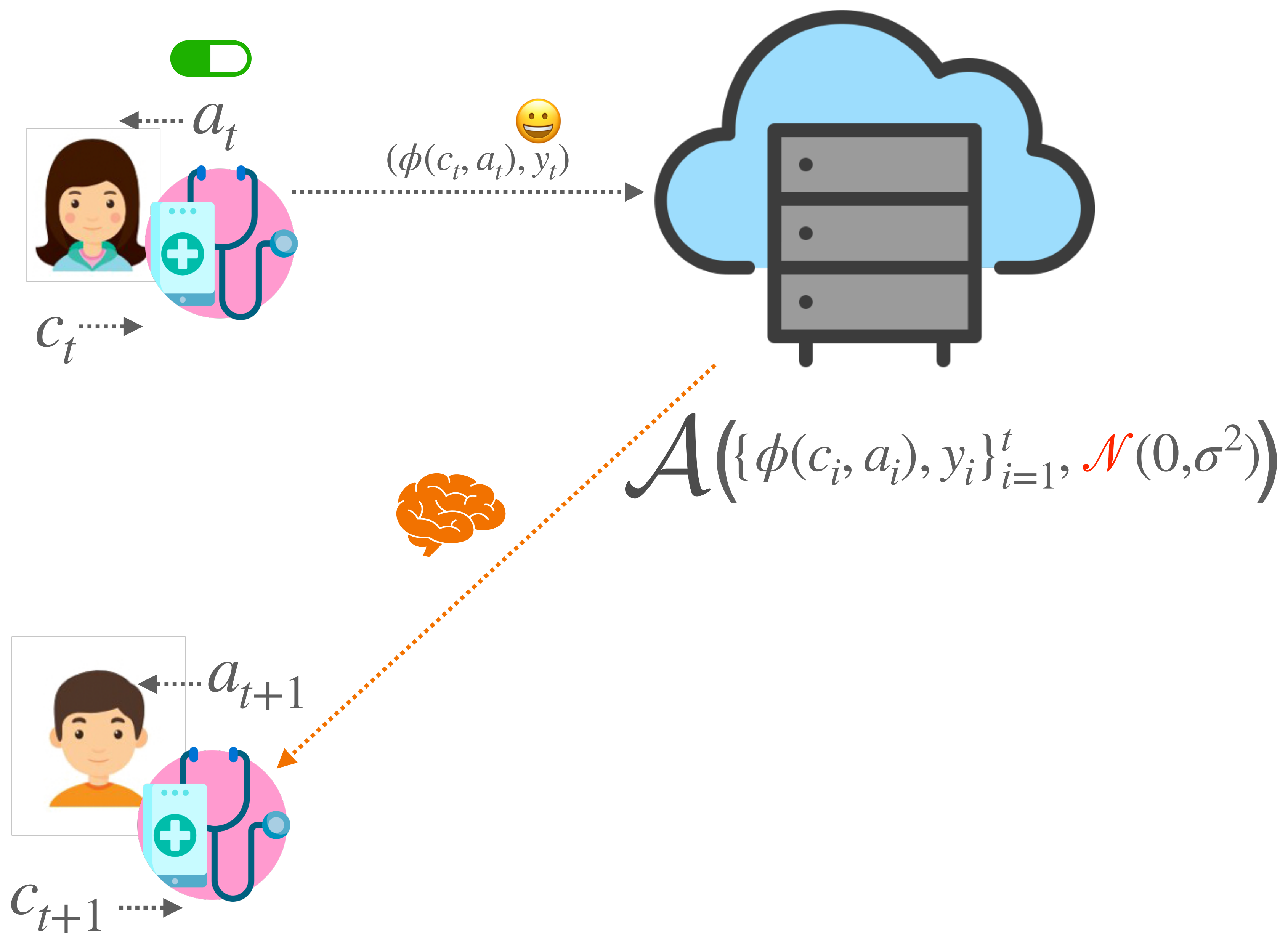
- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise



Differentially Private LCB

Central model

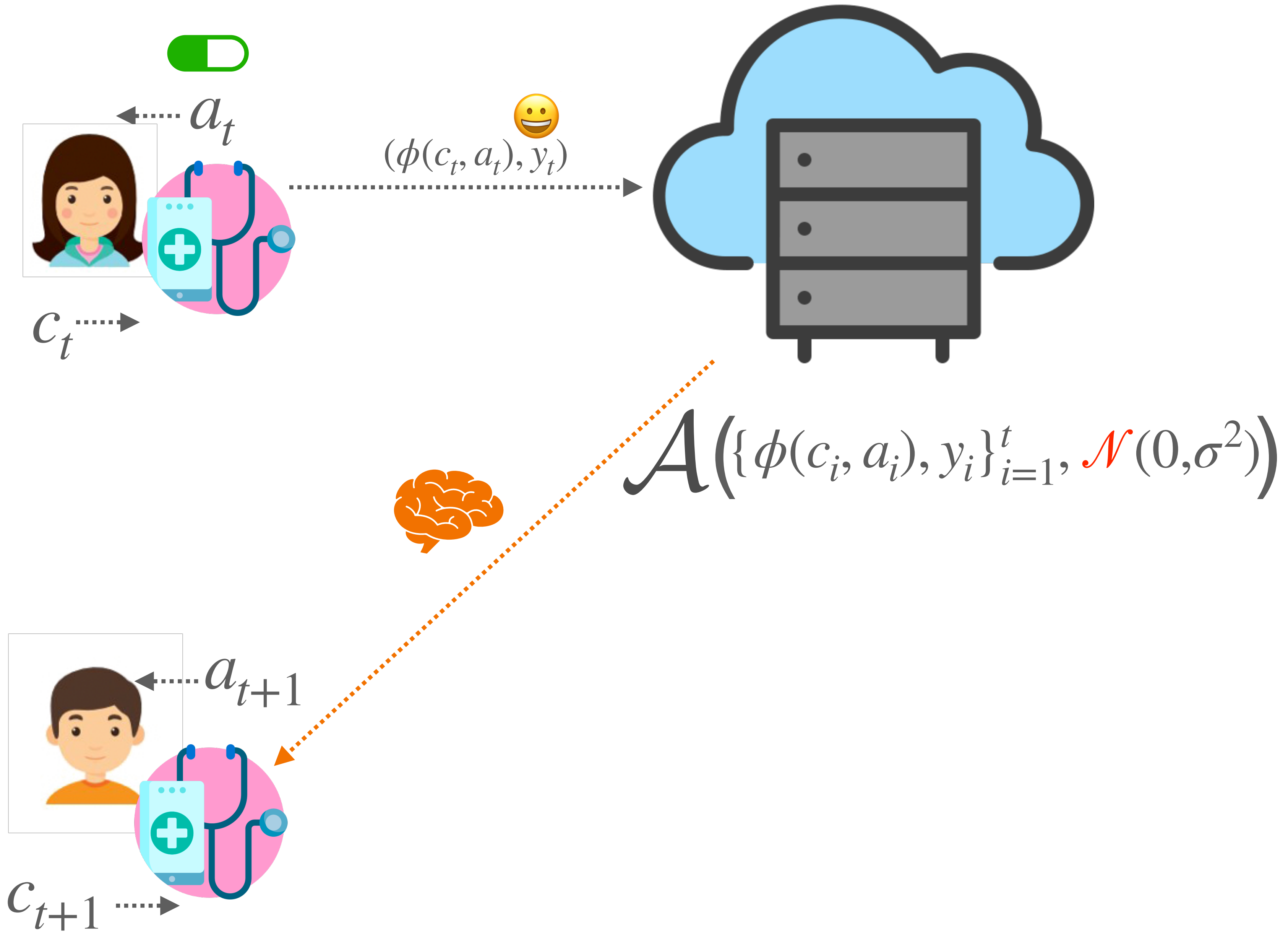
- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$



Differentially Private LCB

Central model

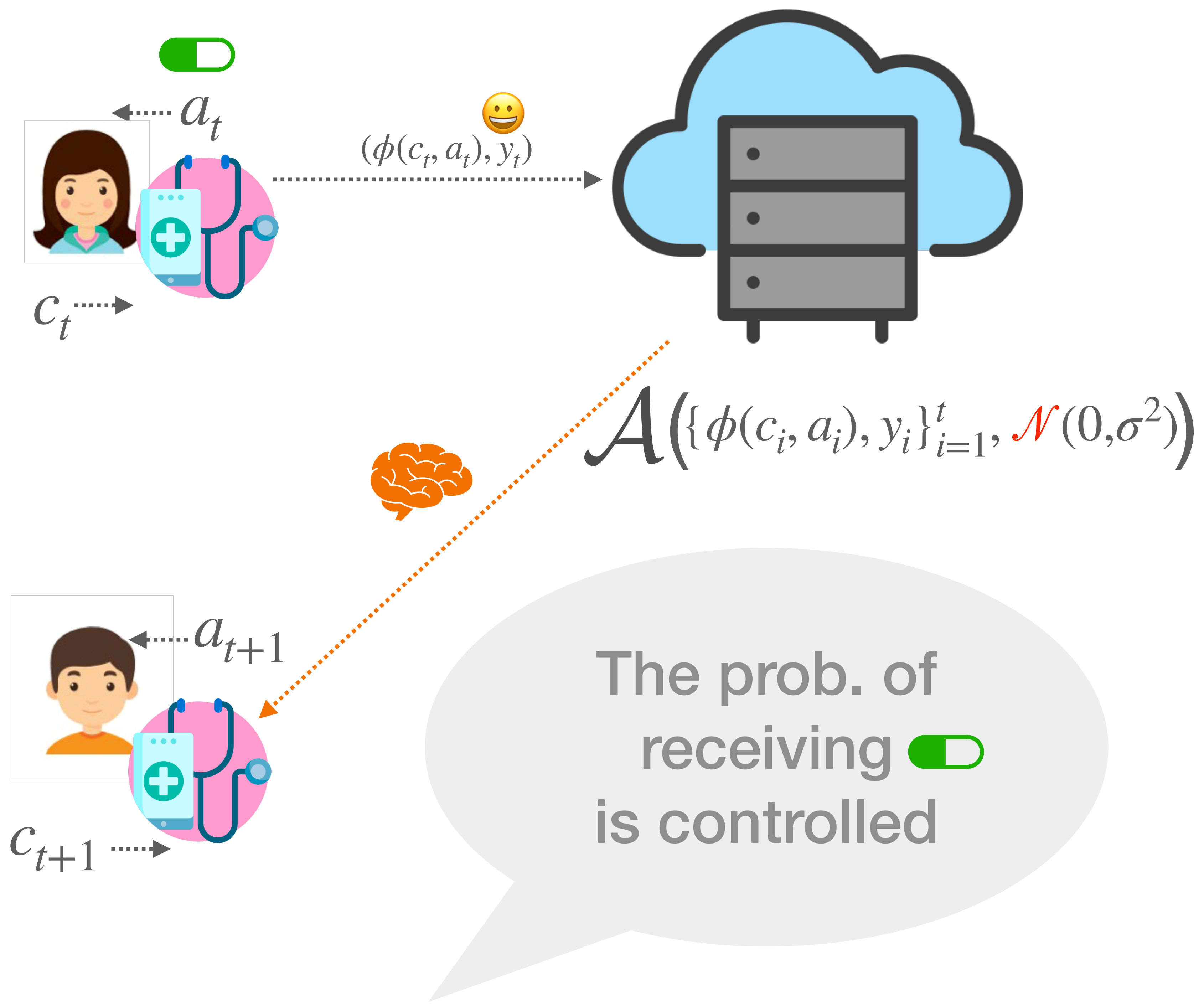
- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$
  - Smaller ϵ, δ, stronger privacy but worse regret



Differentially Private LCB

Central model

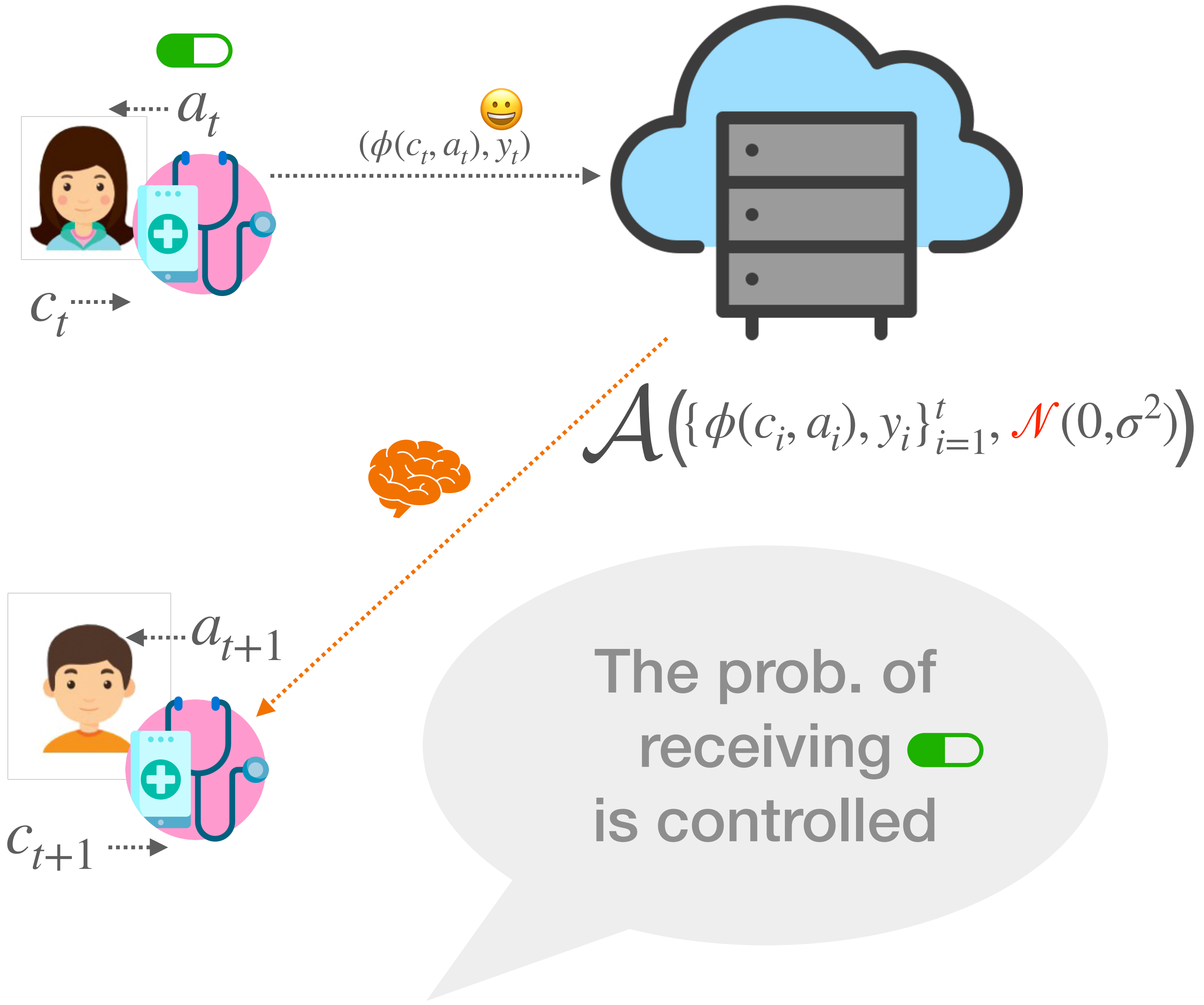
- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$
  - Smaller ϵ, δ, stronger privacy but worse regret



Differentially Private LCB

Central model

- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$
  - Smaller ϵ, δ, stronger privacy but worse regret
- **Privacy vs Regret.** [Shariff and Sheffet. 2018] shows that

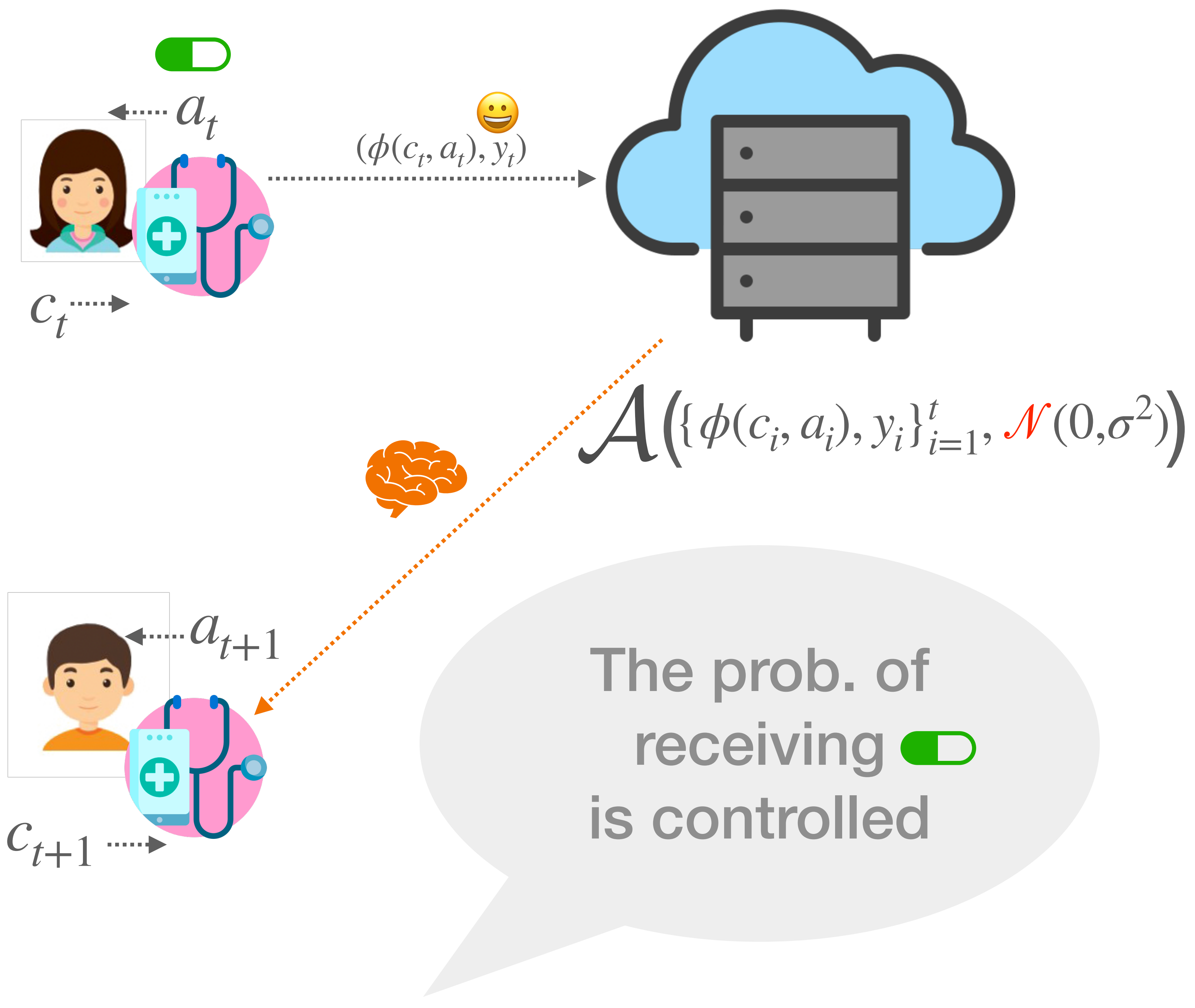


Differentially Private LCB

Central model

- Differential Privacy (DP) provides formal privacy guarantee [Dwork et al. 2006]
- Well-tuned noise added to obscure each user's contribution
- In LCB, **central server** updates model with injected noise
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$
  - Smaller ϵ, δ, stronger privacy but worse regret
- **Privacy vs Regret.** [Shariff and Sheffet. 2018] shows that

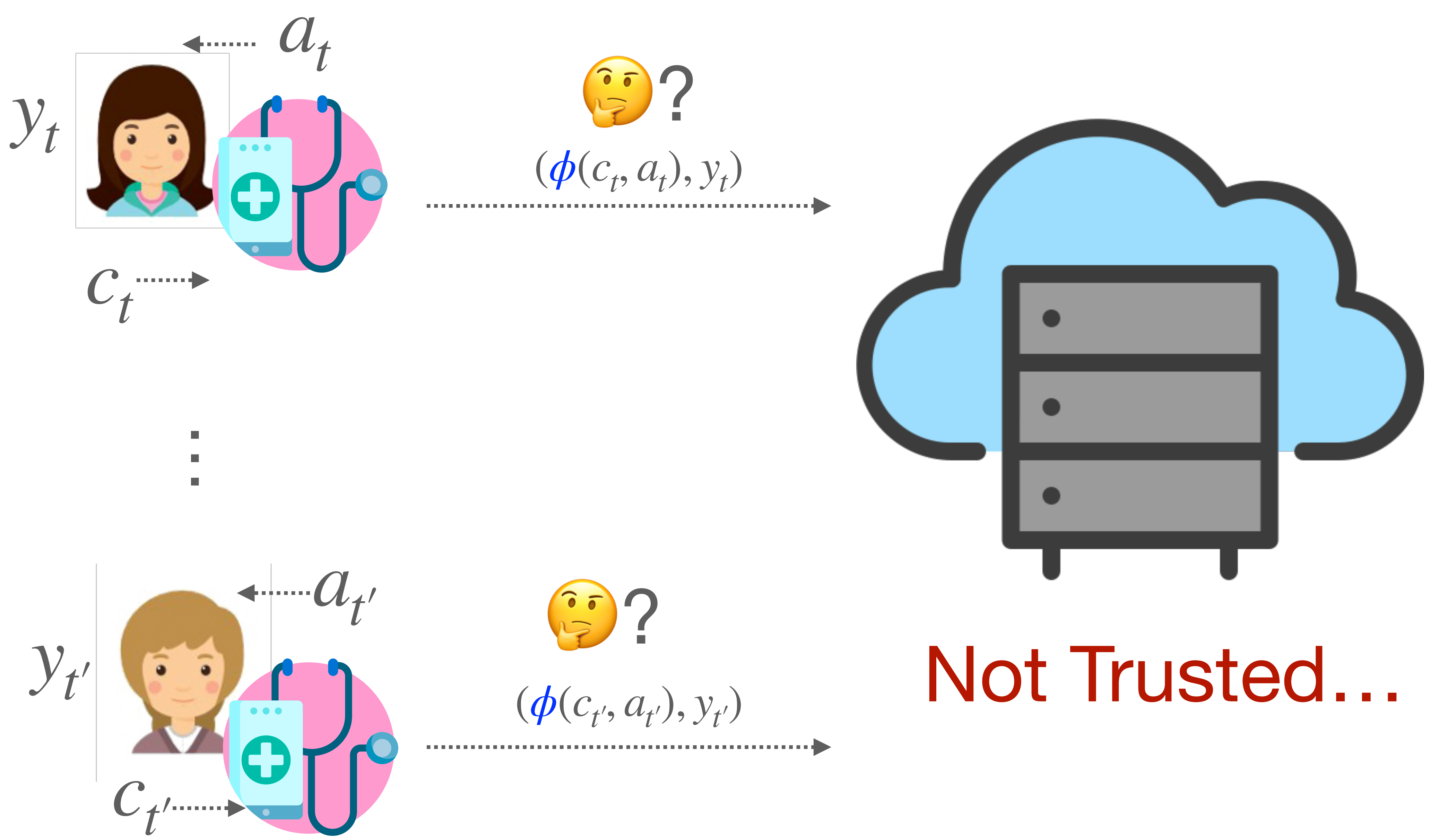
Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP*



*Contextual bandits need relax to joint-DP

Another Privacy Risk

- Both **context** and **reward** are sensitive information
- What if central server is not trustworthy?
  - *Will it follow the right DP mechanism…?*
  - *Will it use my data for other use cases…?*
  - *Will it be attacked by an adversary…?*
- **Hence**, users may not be willing to share their raw data
  - Context via $\phi(c_t, a_t)$
  - Reward y_t

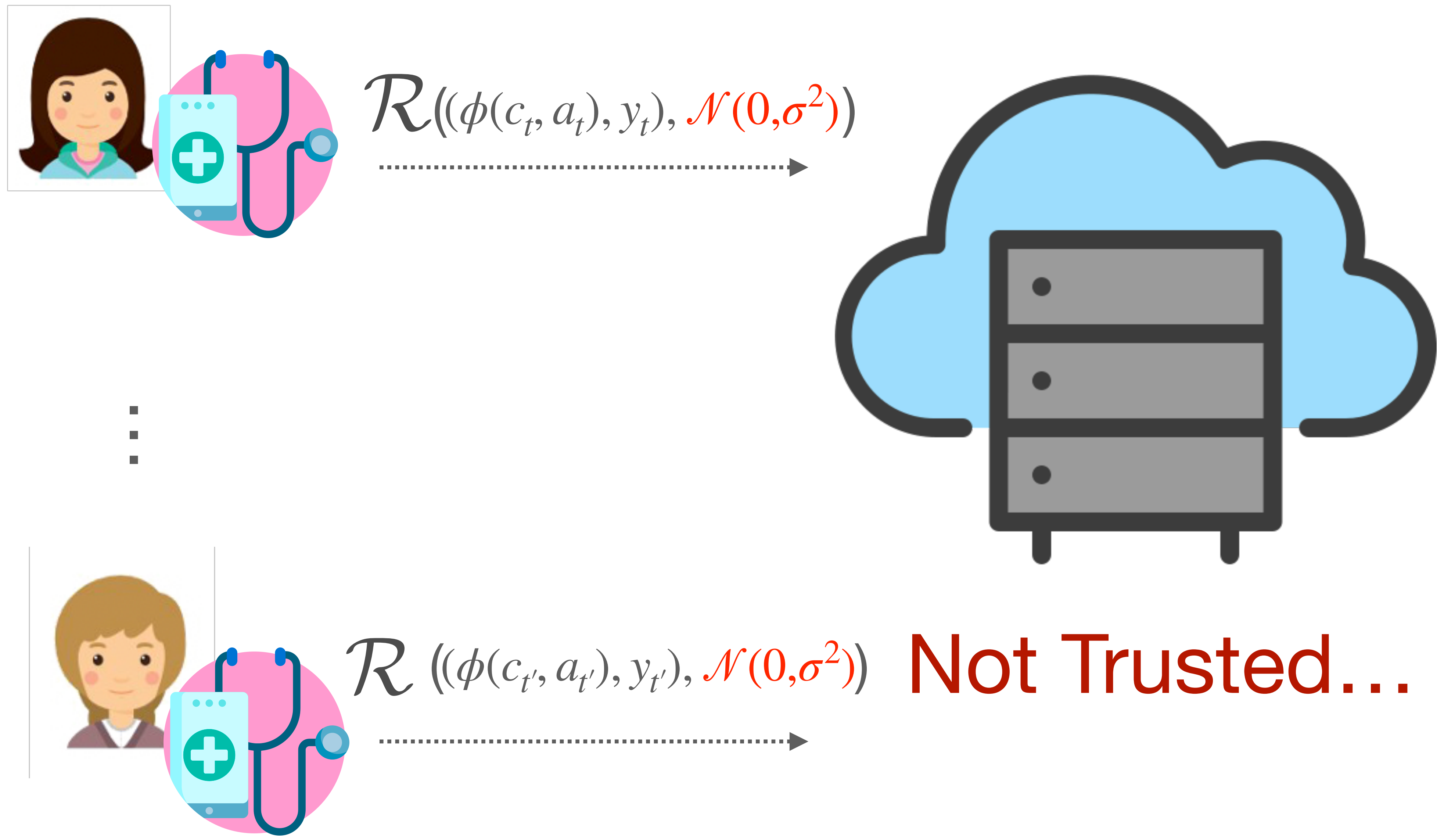


Differentially Private LCB

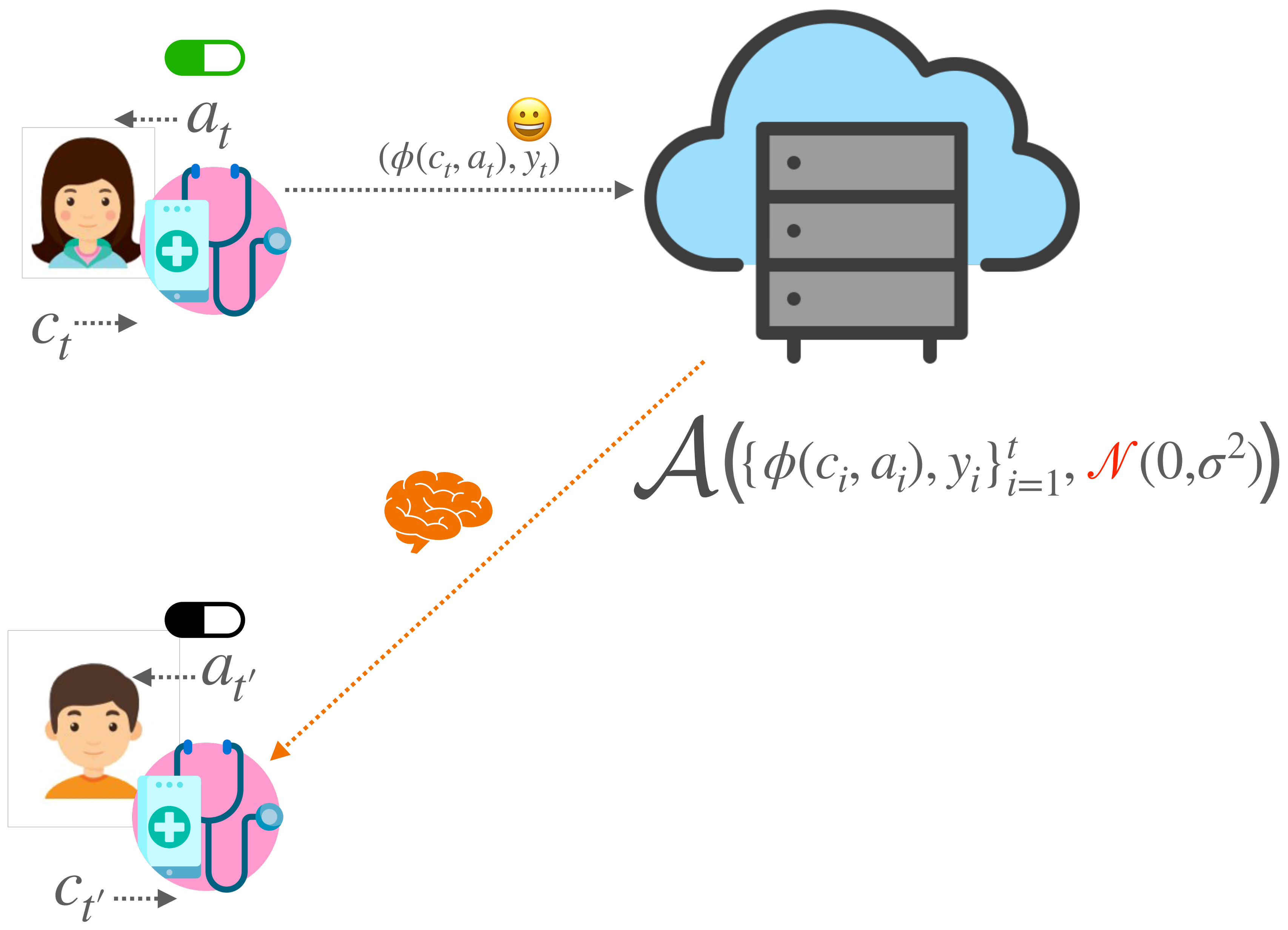
Local model

- Each user injects noise before sending data
  - By post-processing, local DP implies central DP
- In LCB, **each user** applies local randomizer $\mathcal{R}$
  - Gaussian noise with variance $\sigma^2 = O(\log(1/\delta)/\epsilon^2)$
  - Smaller ϵ, δ, stronger privacy but worse regret
- **Privacy vs Regret.** [Zheng et al. 2020] shows that

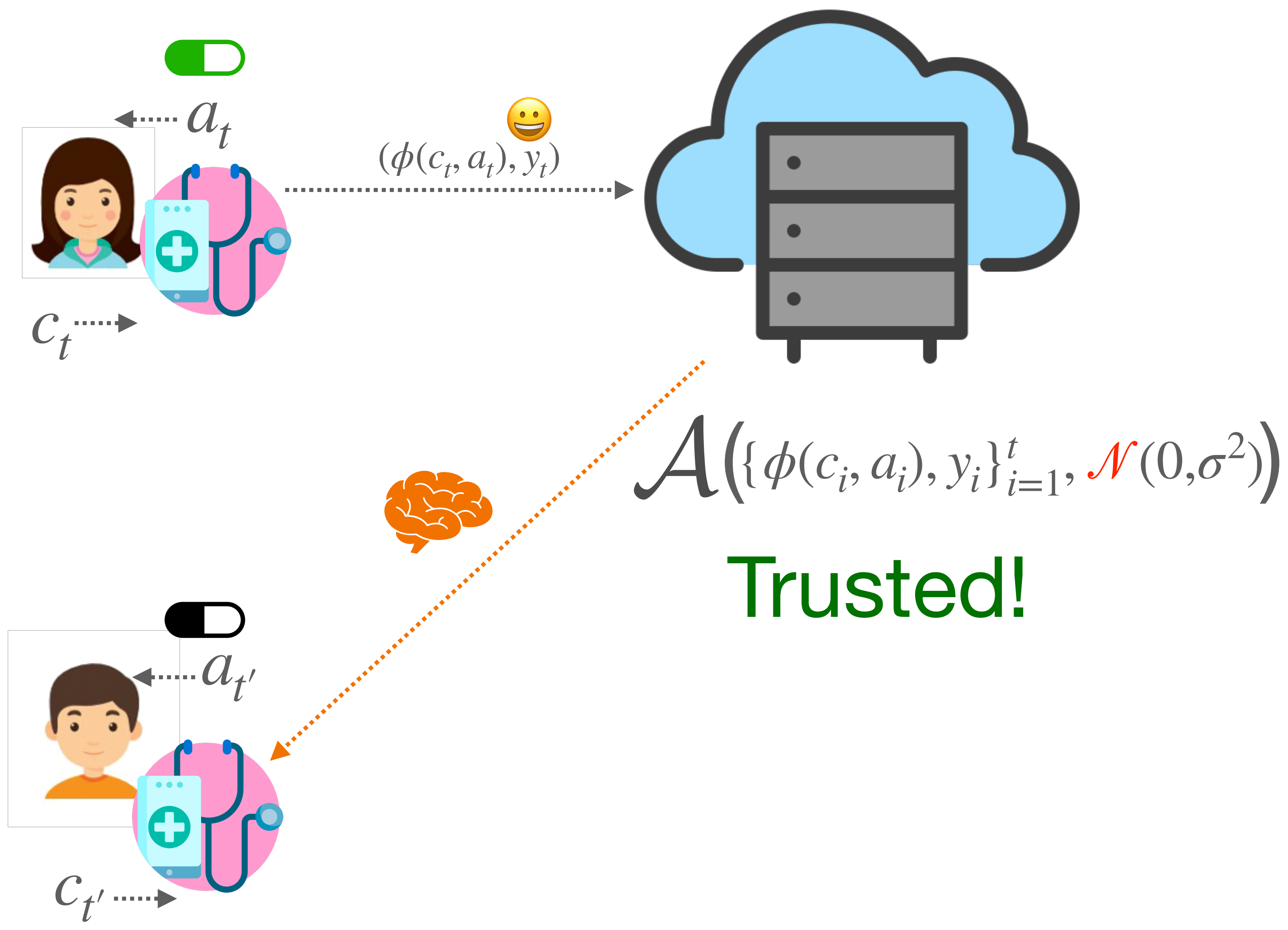
Regret $\tilde{O}\left(\frac{T^{3/4}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under local (ϵ, δ)-DP*



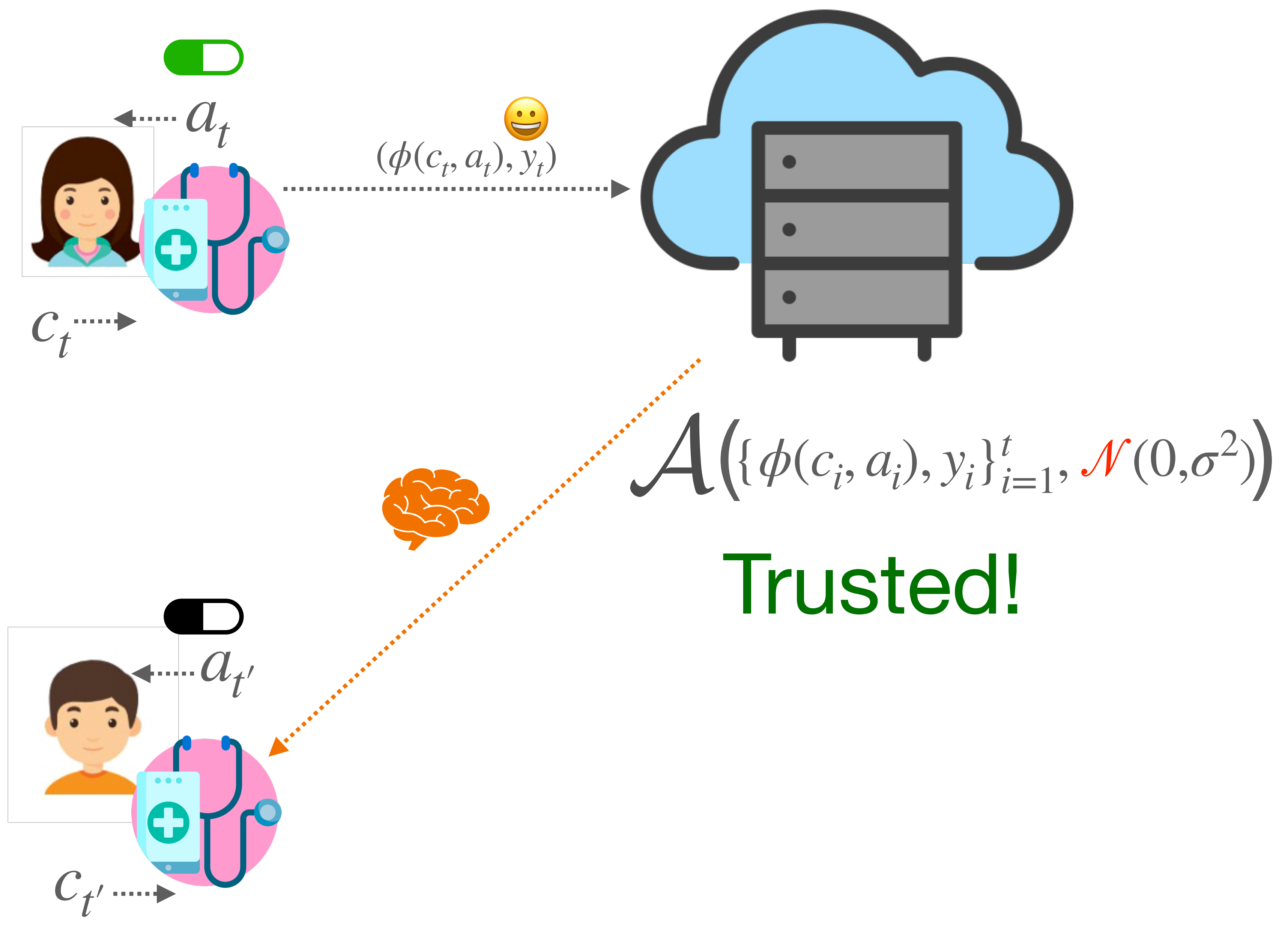
* The original result is $1/\epsilon$, which can be easily improved to $1/\sqrt{\epsilon}$



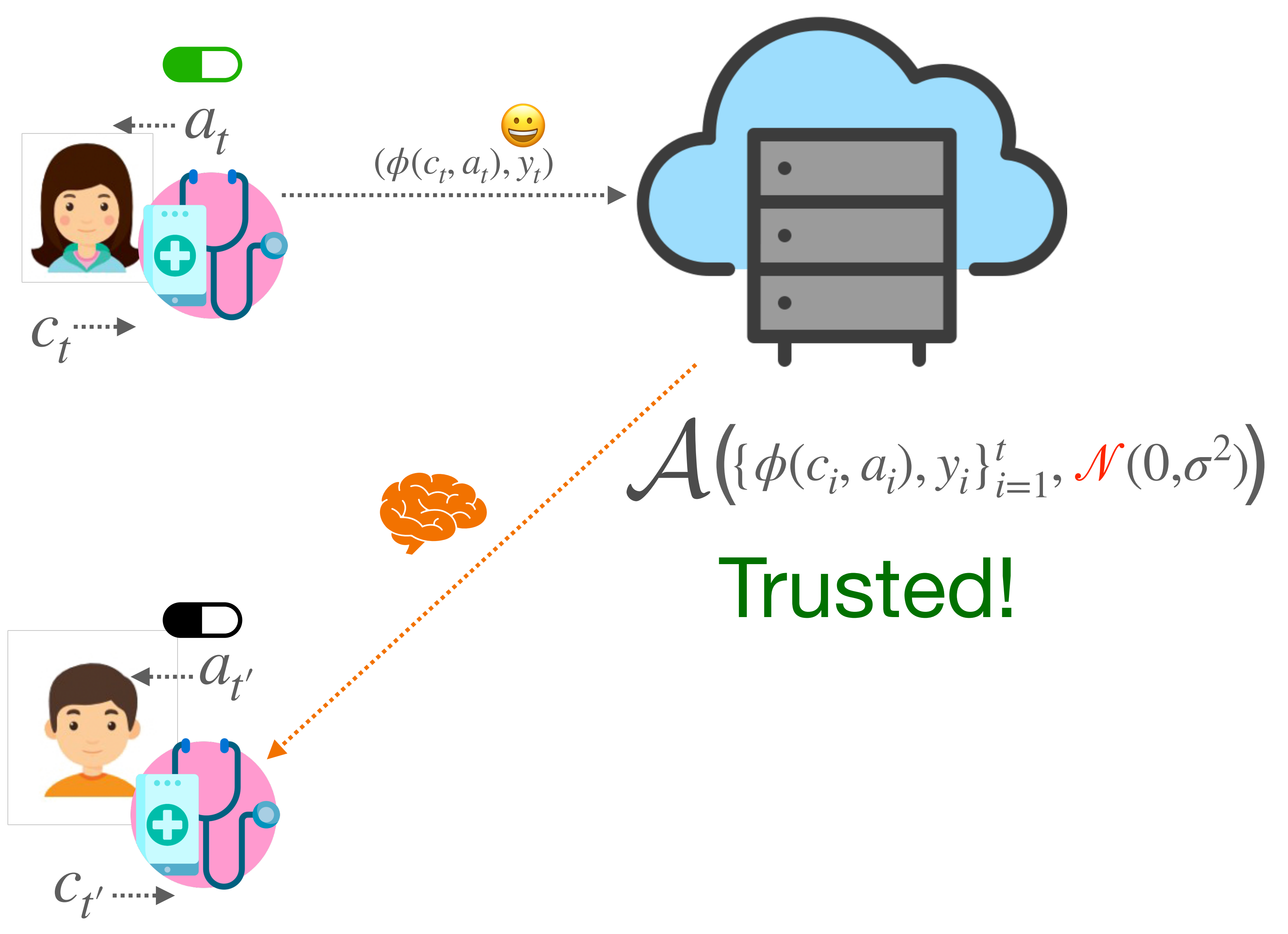
a_t
$(\phi(c_t, a_t), y_t)$
c_t
$\mathcal{A}\left(\{\phi(c_i, a_i), y_i\}_{i=1}^{t}, \mathcal{N}(0,\sigma^2)\right)$
$a_{t'}$
$c_{t'}$

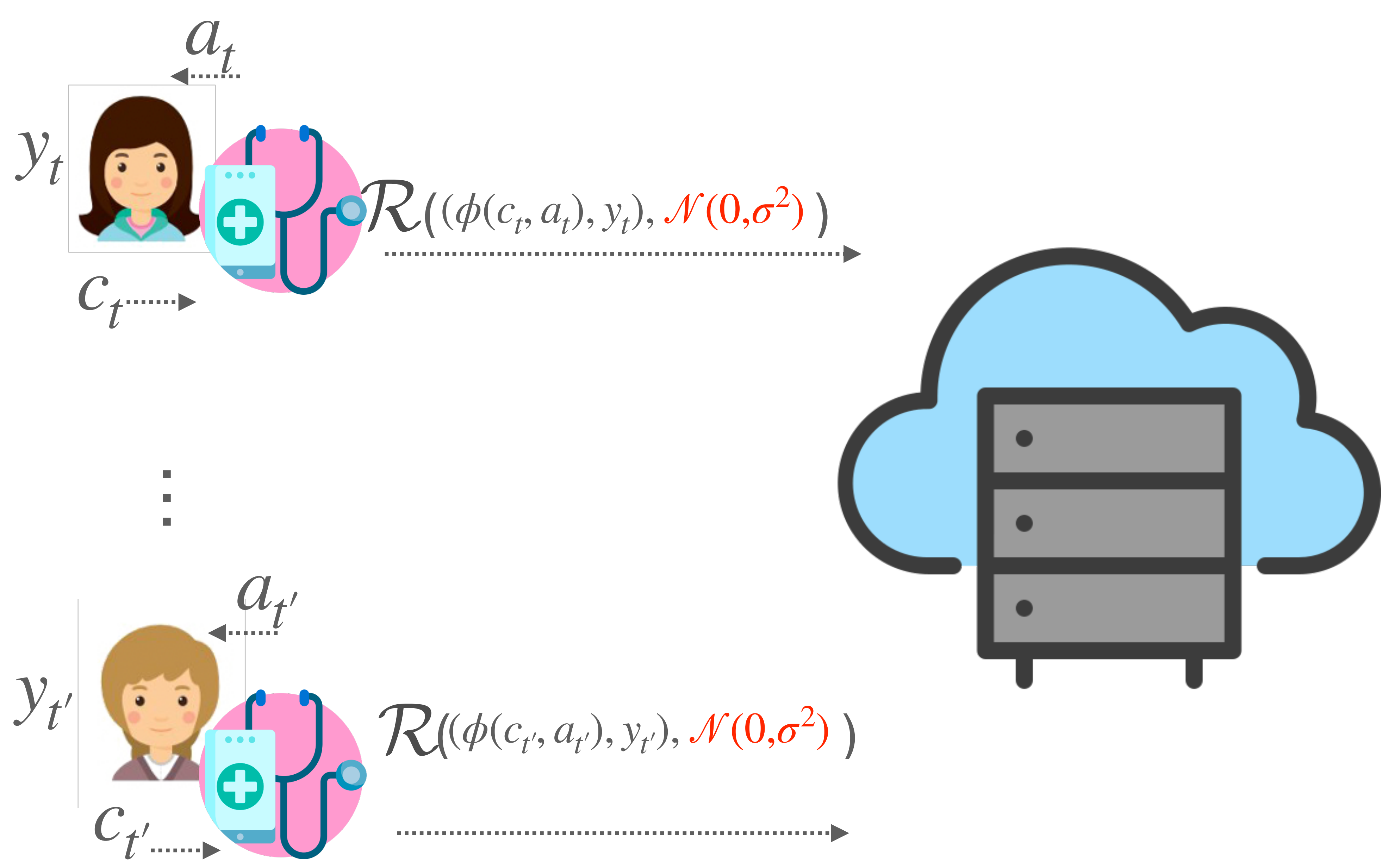


a_t
$(\phi(c_t, a_t), y_t)$
c_t
$\mathcal{A}\left(\{\phi(c_i, a_i), y_i\}_{i=1}^t, \mathcal{N}(0,\sigma^2)\right)$
Trusted!
$a_{t'}$
$c_{t'}$

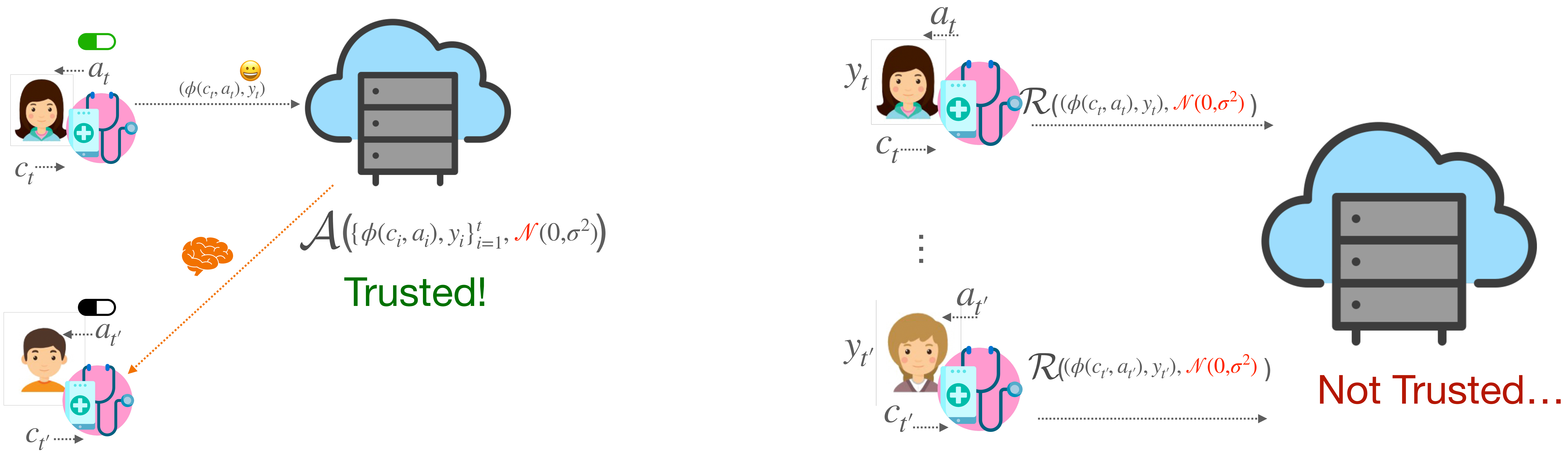


Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP

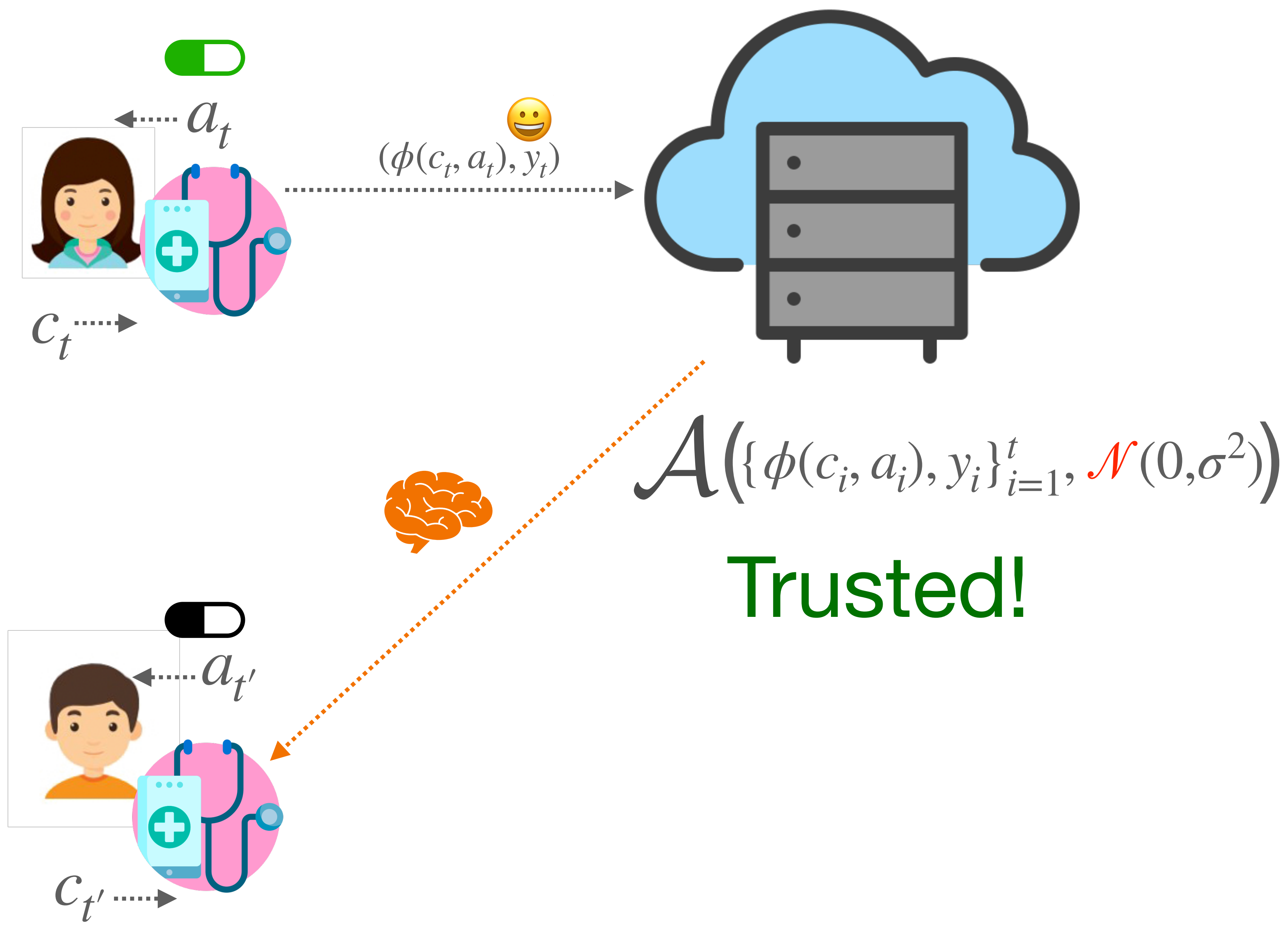




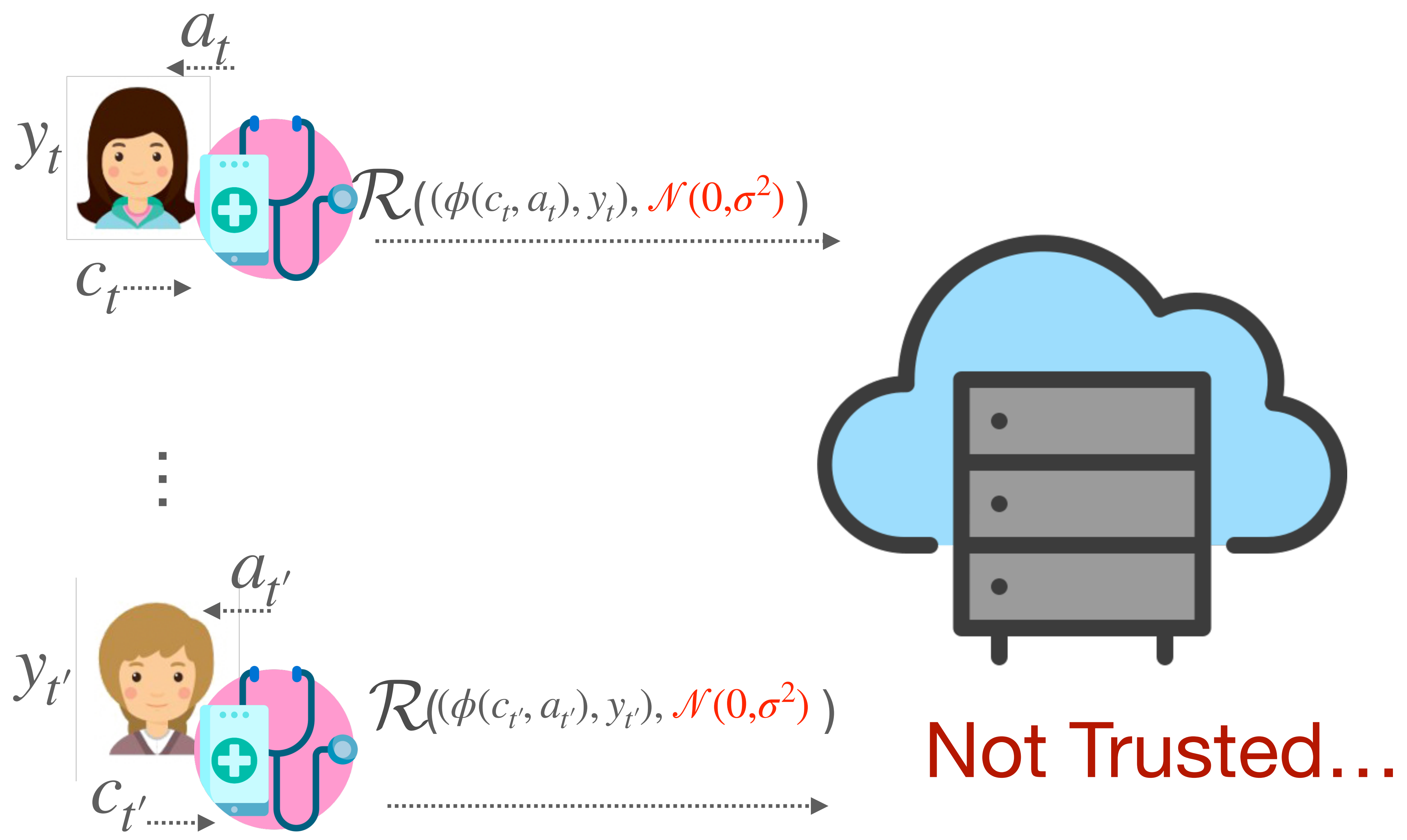
Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP



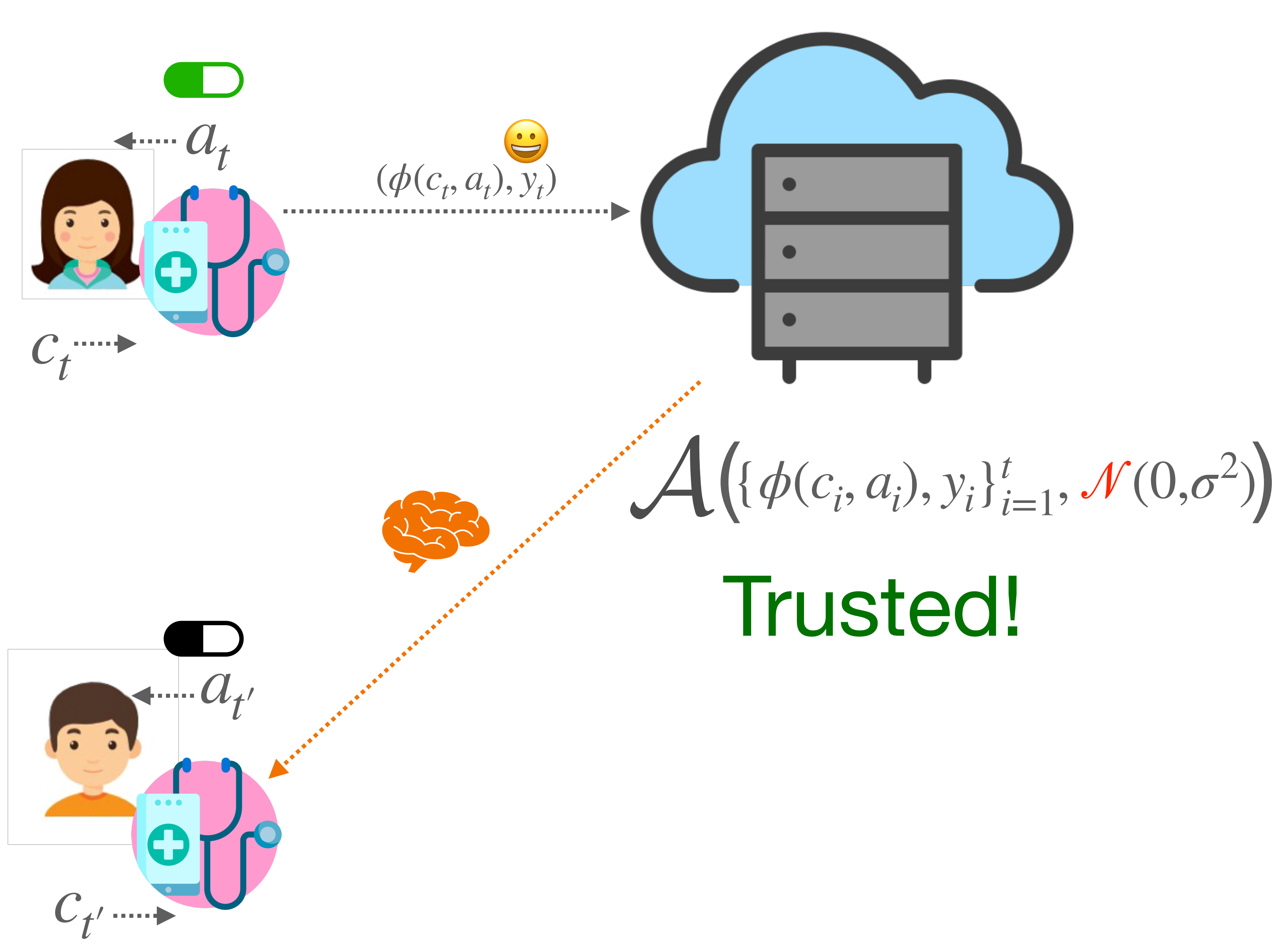
Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP

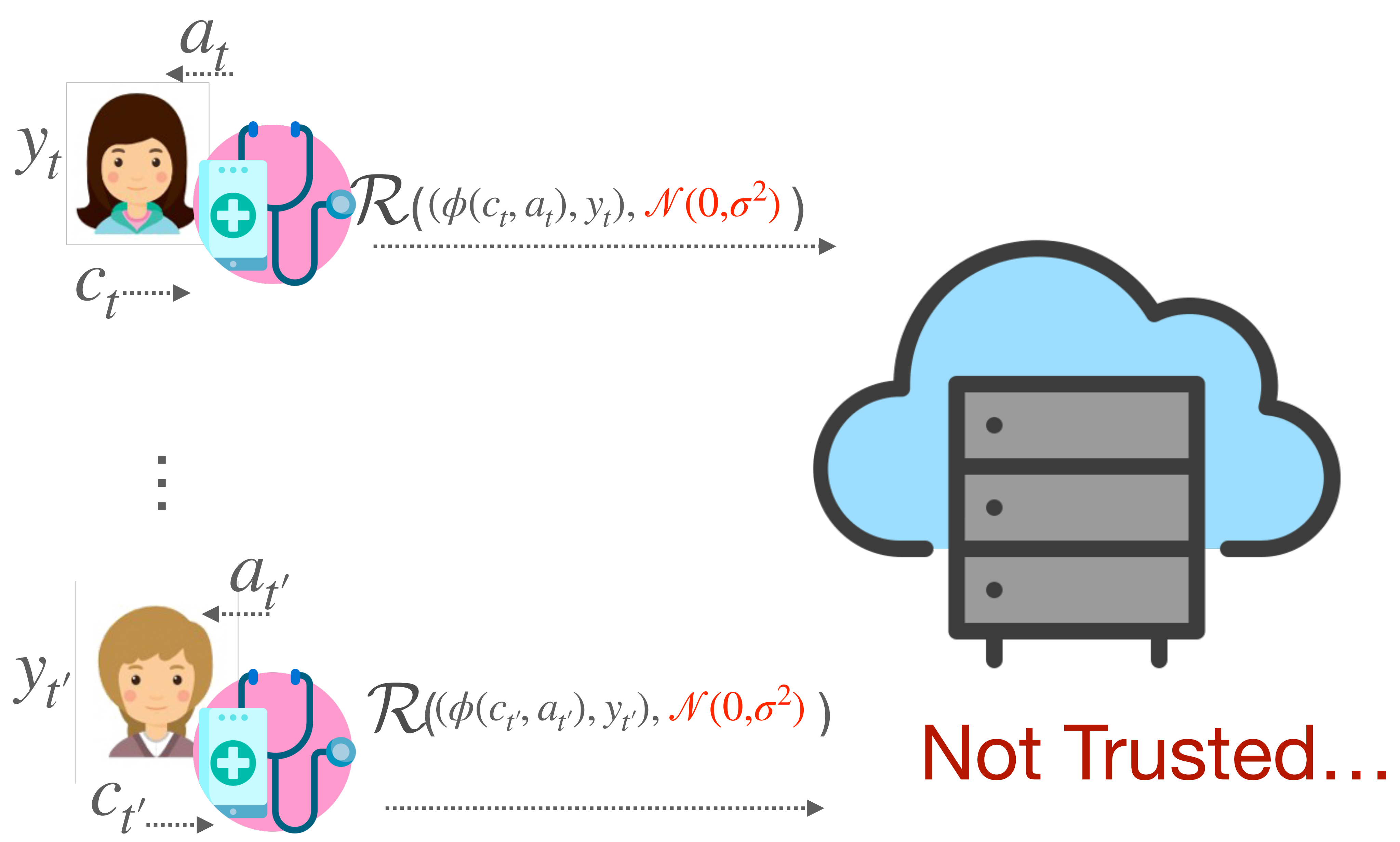


Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP



Regret $\tilde{O}\left(\frac{T^{3/4}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under local (ϵ, δ)-DP

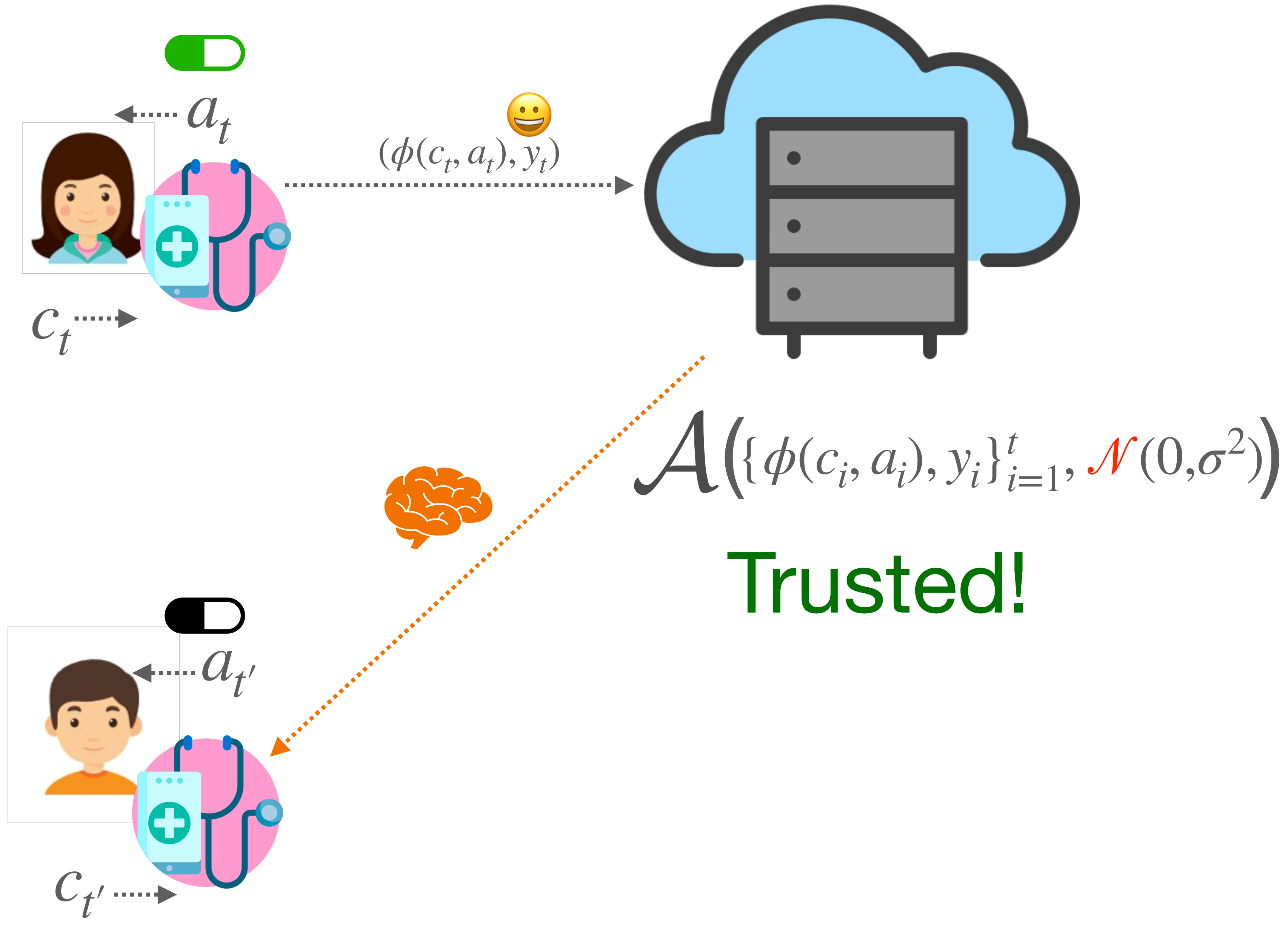


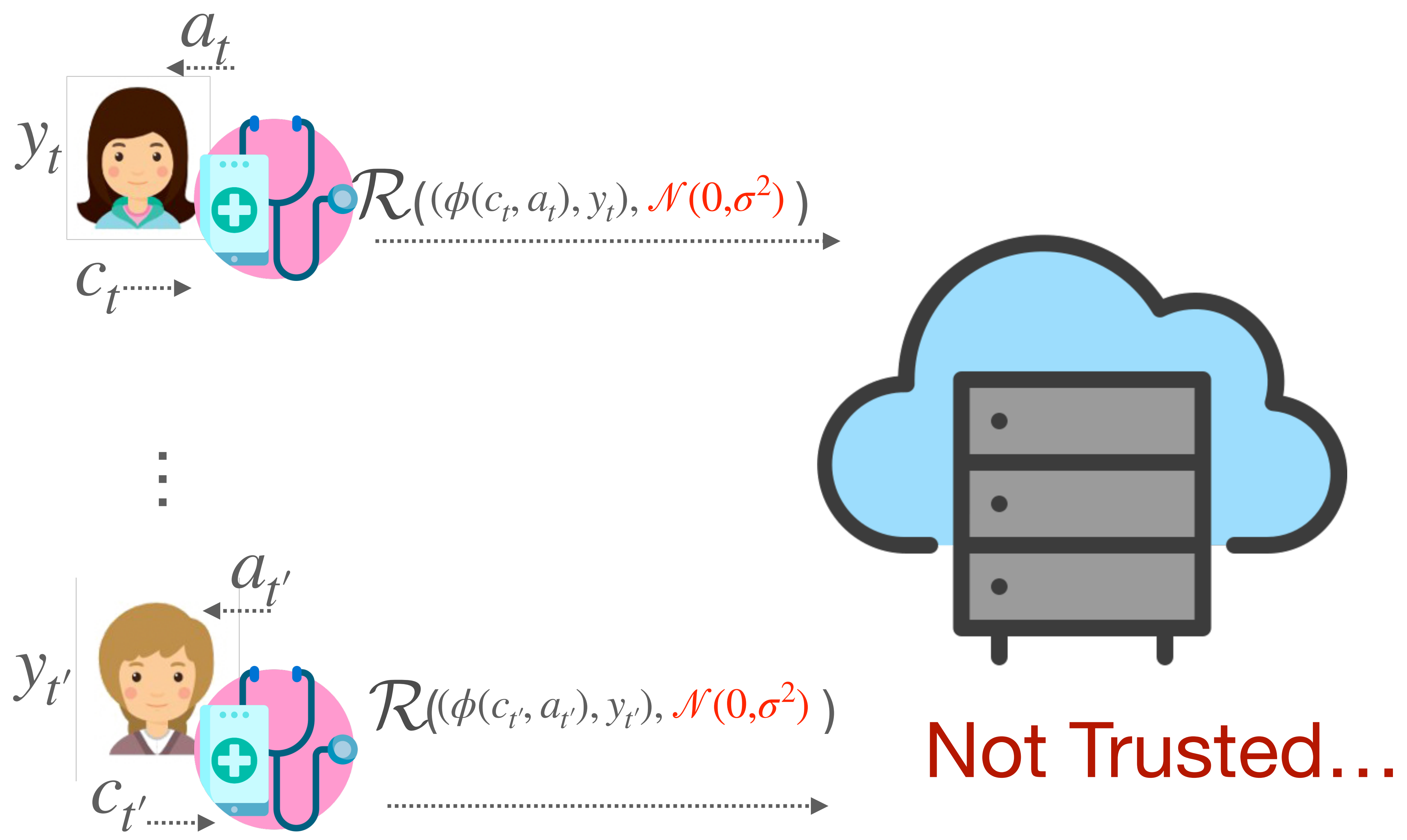


Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ **under central** (ϵ, δ)**-DP**

Regret $\tilde{O}\left(\frac{T^{3/4}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ **under local** (ϵ, δ)**-DP**

Can one achieve a better regret even without a trusted server?





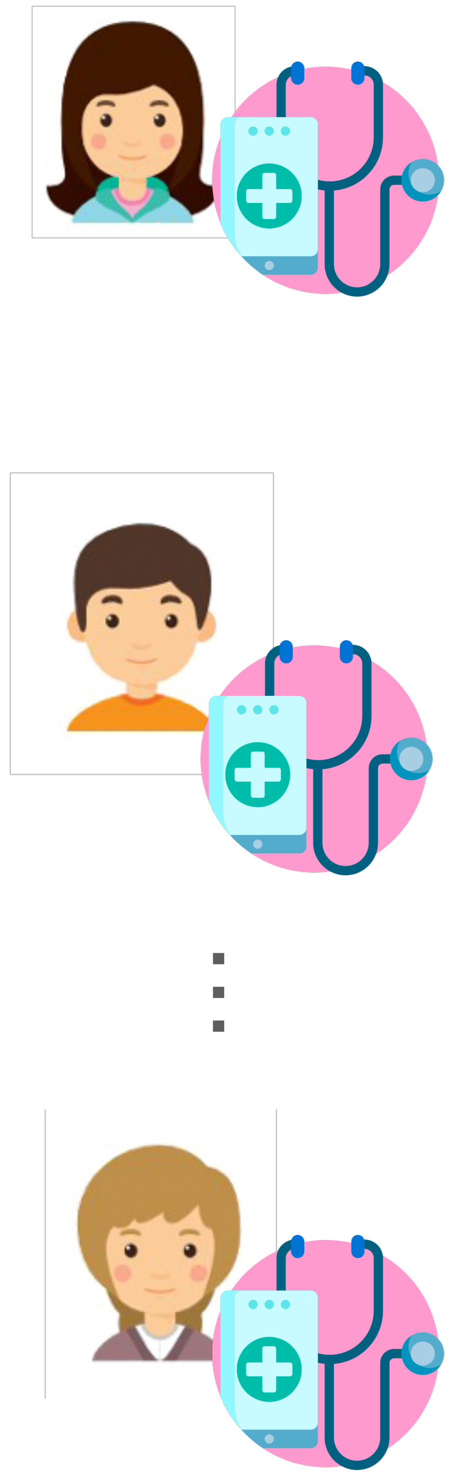
Regret $\tilde{O}\left(\frac{\sqrt{T}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under central (ϵ, δ)-DP

Regret $\tilde{O}\left(\frac{T^{3/4}(\log(1/\delta))^{1/4}}{\sqrt{\epsilon}}\right)$ under local (ϵ, δ)-DP

Can one achieve a better regret even without a trusted server?

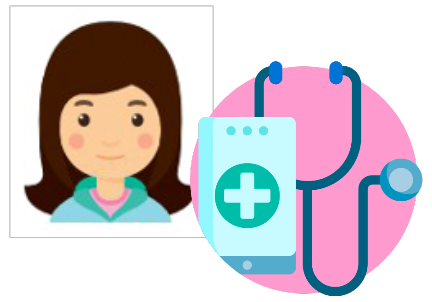
Yes!

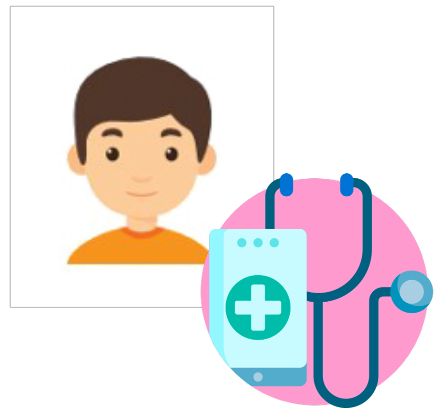
Contribution



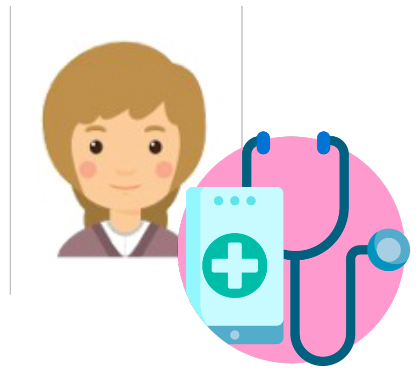
$\mathcal{A}$

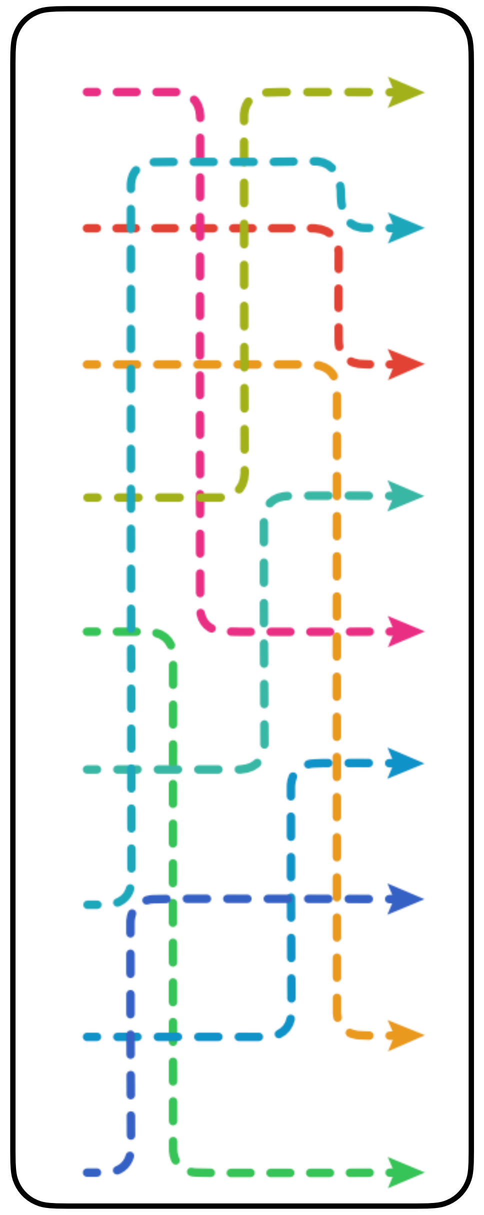
Not Trusted…

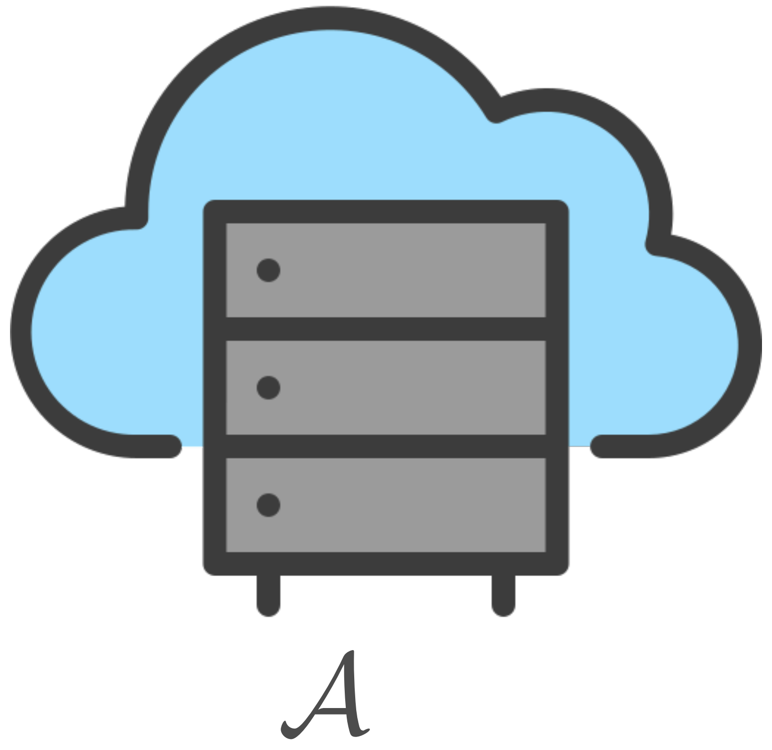




⋮

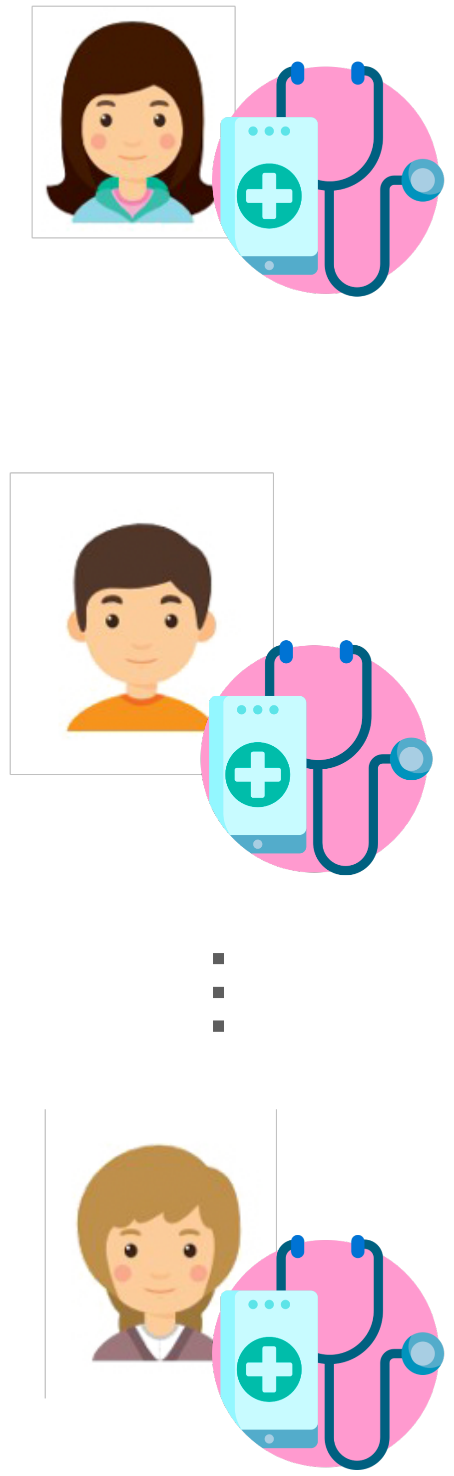


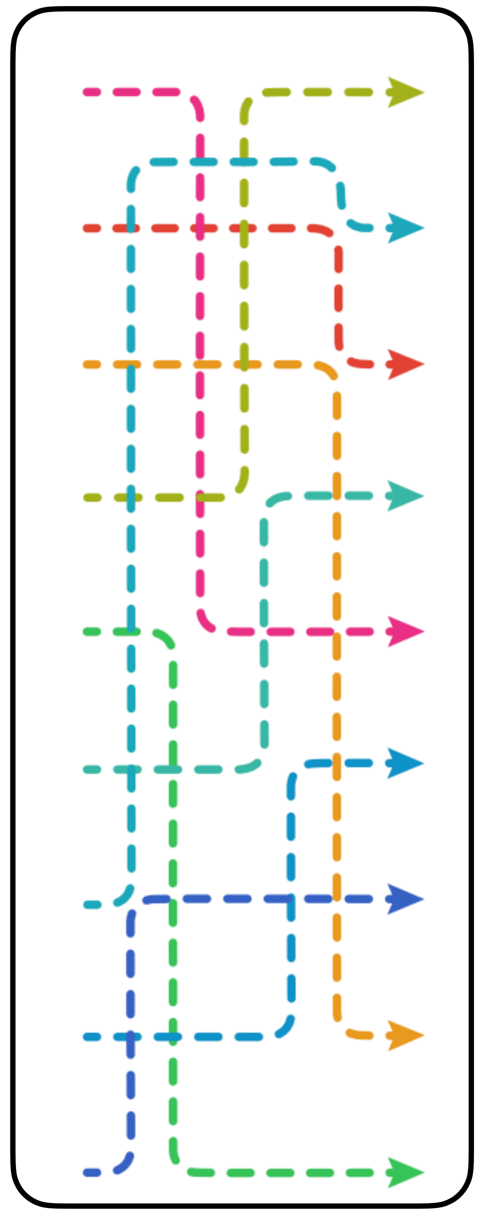




Not Trusted…

*Shuffler's figure borrows from [Cheu et al. 2019]





Shuffler: $\mathcal{S}$

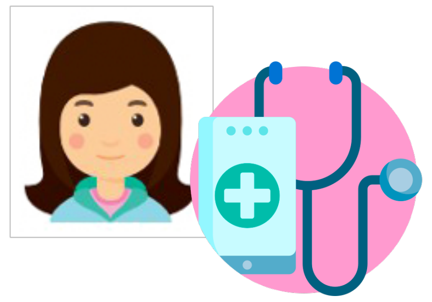
additional randomness “amplify” privacy

$\mathcal{A}$

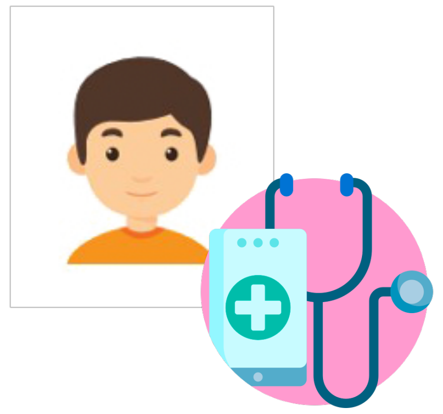
Not Trusted…

*Shuffler’s figure borrows from [Cheu et al. 2019]

output is (ϵ_0, δ_0) **private**, $\epsilon_0 \gg \epsilon, \delta_0 \gg \delta$

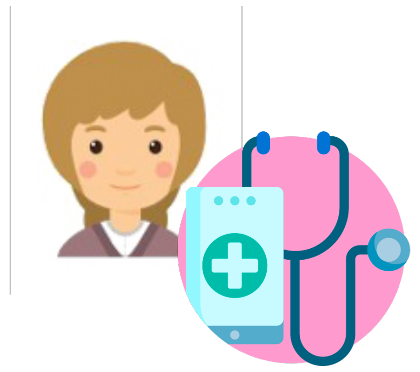


$\mathcal{R}((\phi(c_t, a_t), y_t), \textit{Noise})$

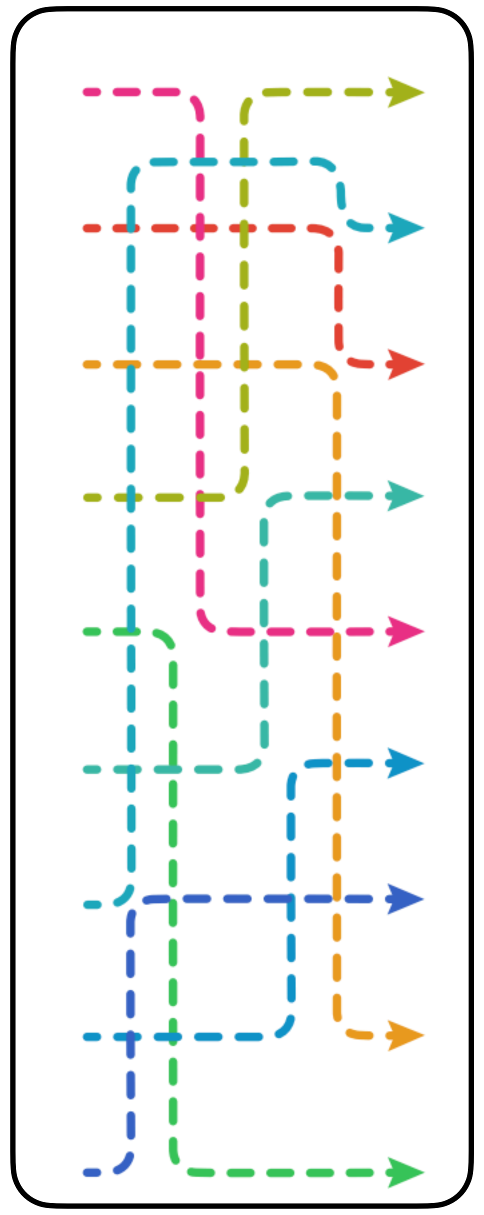


$\mathcal{R}((\phi(c_{t+1}, a_{t+1}), y_{t+1}), \textit{Noise})$

⋮



$\mathcal{R}((\phi(c_{t'}, a_{t'}), y_{t'}), \textit{Noise})$



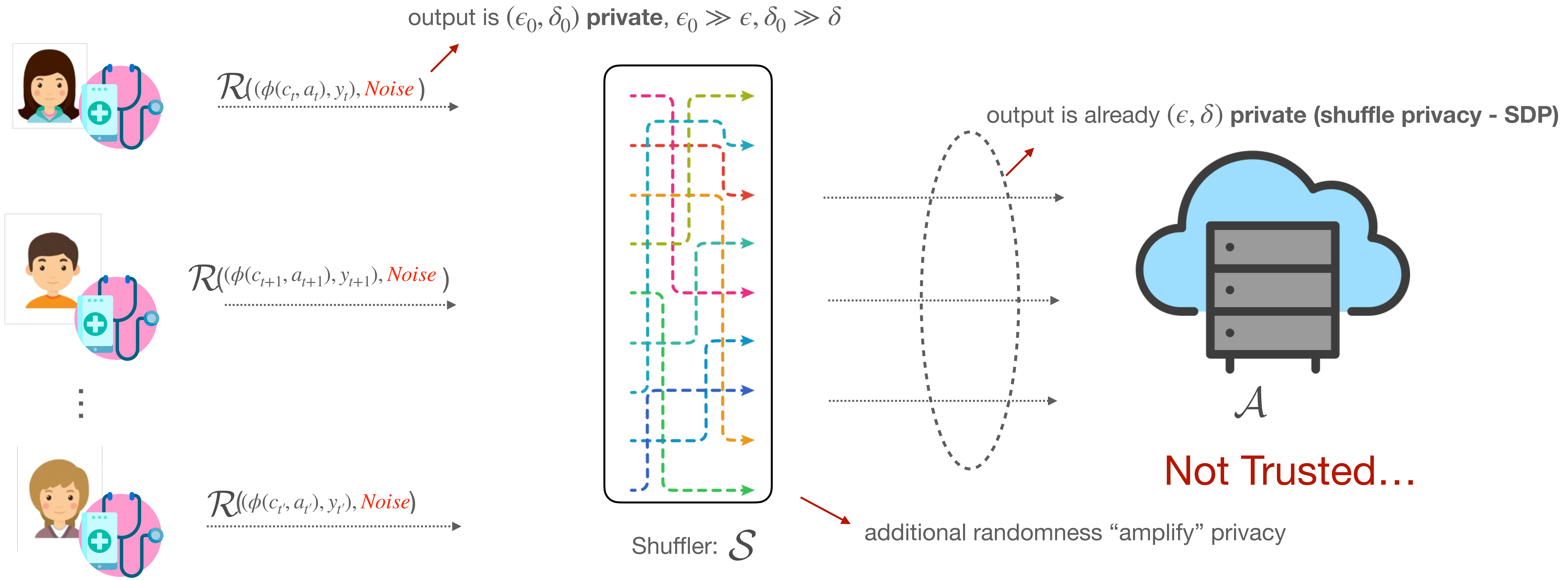
Shuffler: $\mathcal{S}$

additional randomness “amplify” privacy

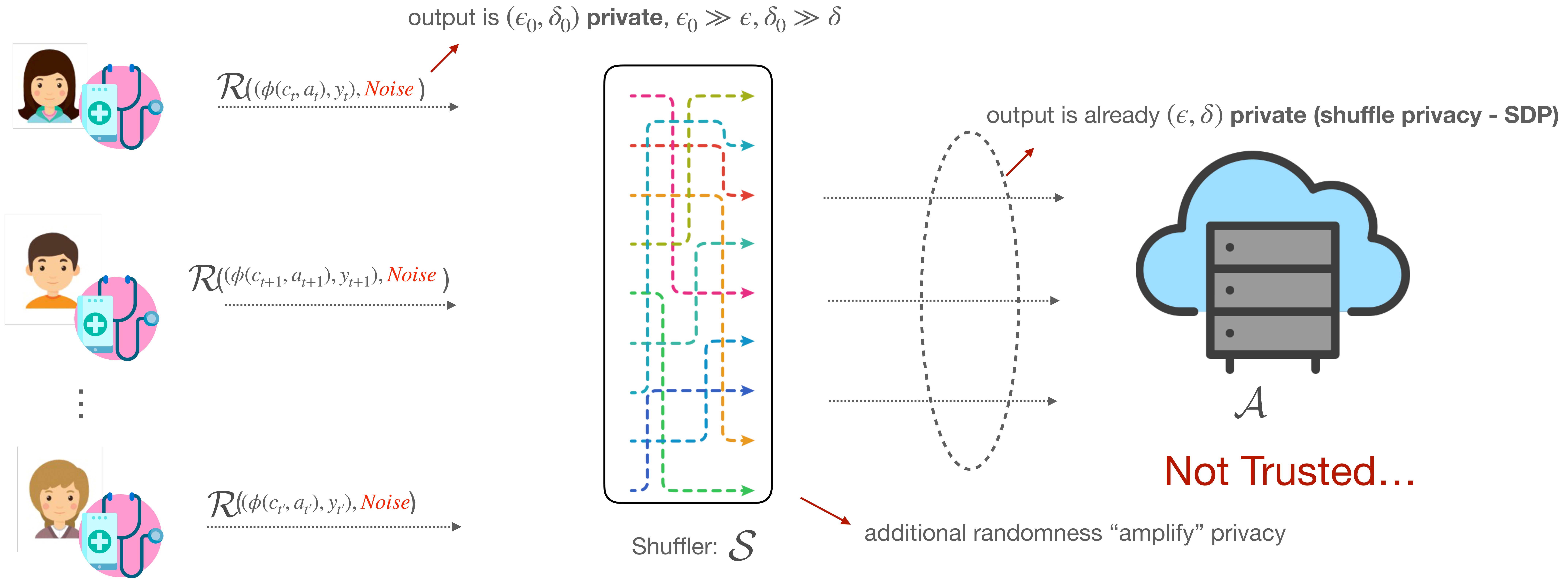
$\mathcal{A}$

Not Trusted…

*Shuffler’s figure borrows from [Cheu et al. 2019]

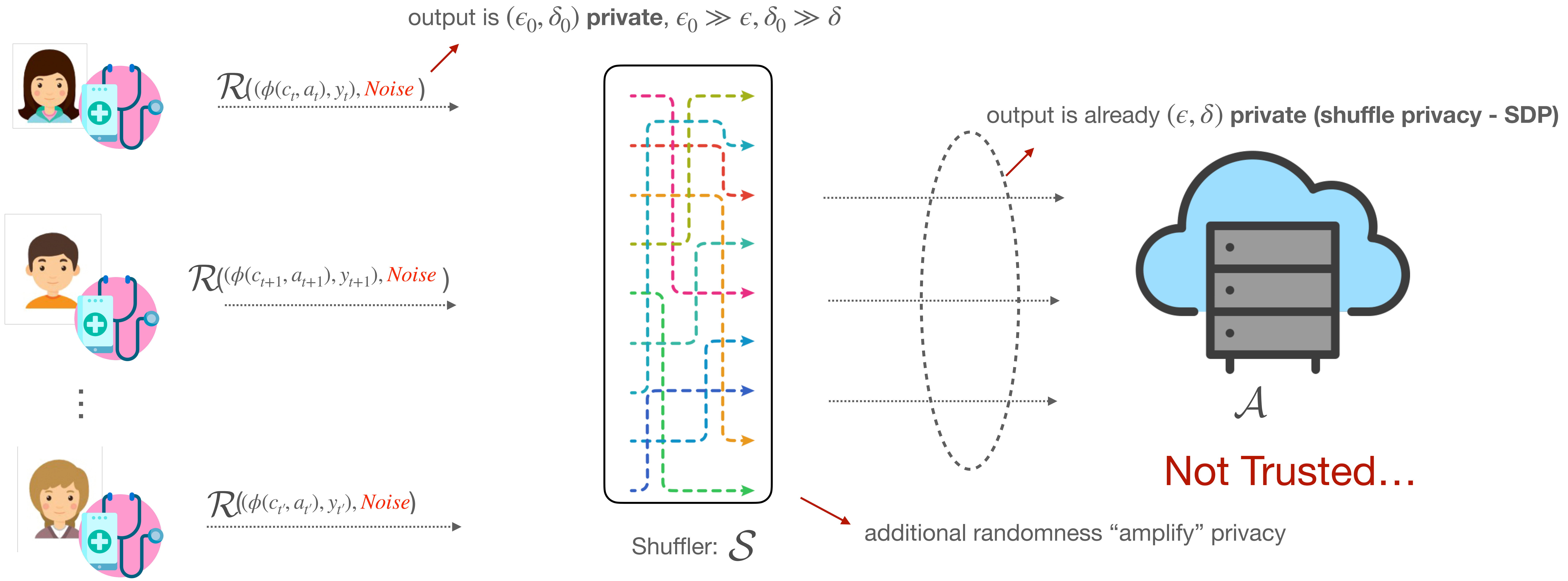


*Shuffler's figure borrows from [Cheu et al. 2019]



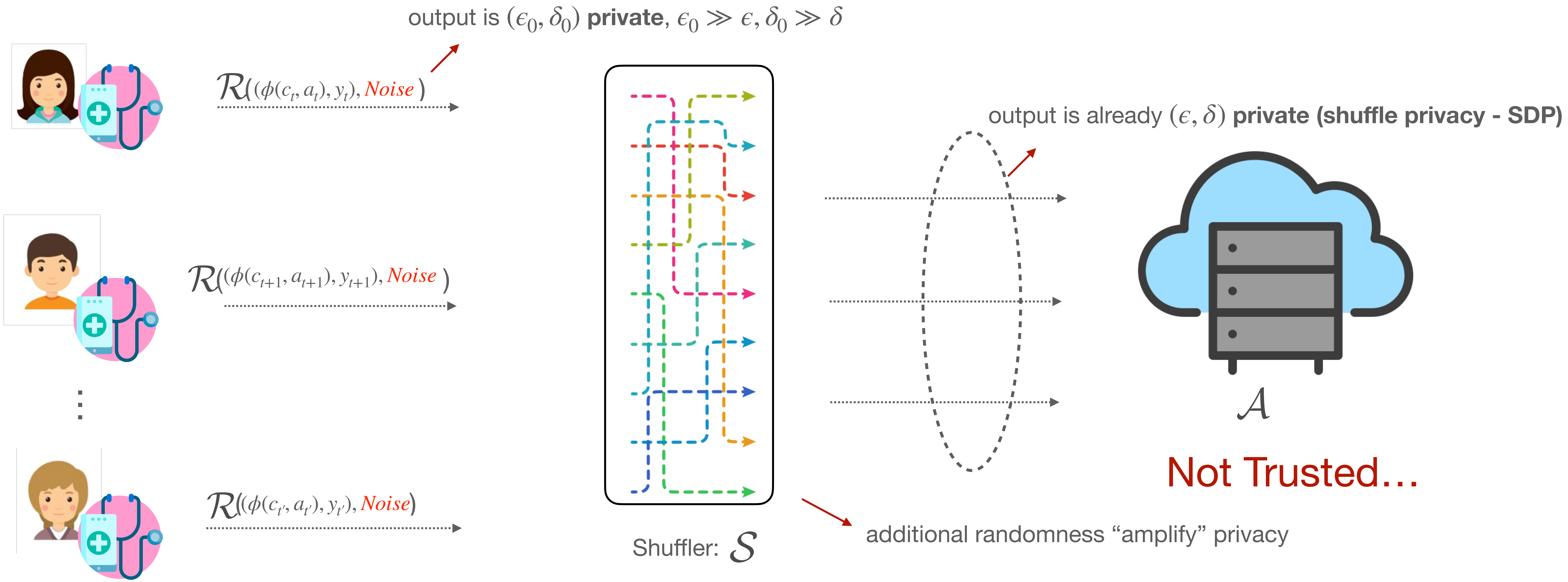
1. Propose a generic private LCB algorithm with black-box protocol $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$

*Shuffler's figure borrows from [Cheu et al. 2019]



1. Propose a generic private LCB algorithm with black-box protocol $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$
2. Two instantiations of $\mathcal{P}$ guarantee *shuffle privacy* with regret $\tilde{O}(T^{3/5})$

*Shuffler’s figure borrows from [Cheu et al. 2019]

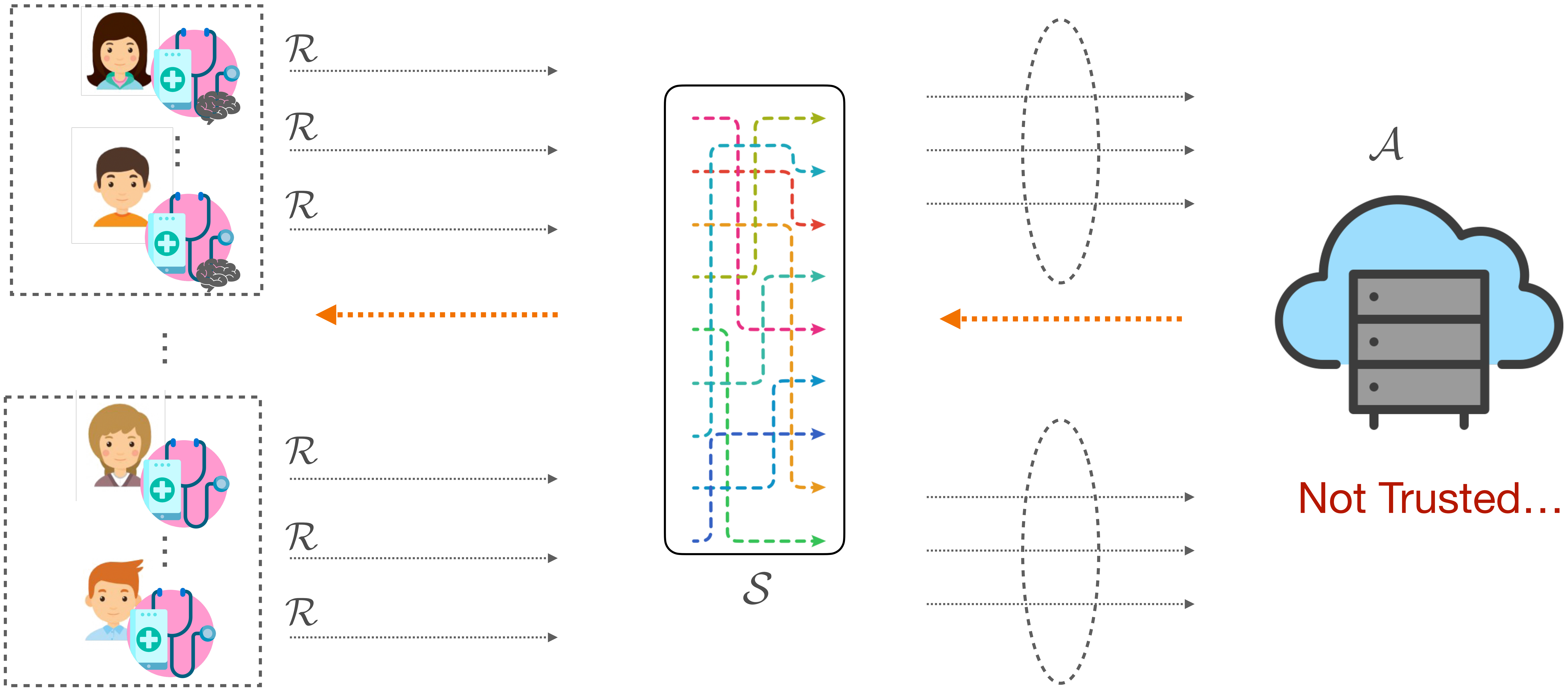


1. Propose a generic private LCB algorithm with black-box protocol $\mathcal{P} = (\mathcal{R}, \mathcal{S}, \mathcal{A})$
2. Two instantiations of $\mathcal{P}$ guarantee *shuffle privacy* with regret $\tilde{O}(T^{3/5})$
3. For the case of returning users, our regret can **match** the one under central model, i.e, $\tilde{O}(T^{2/3})$

*Shuffler’s figure borrows from [Cheu et al. 2019]

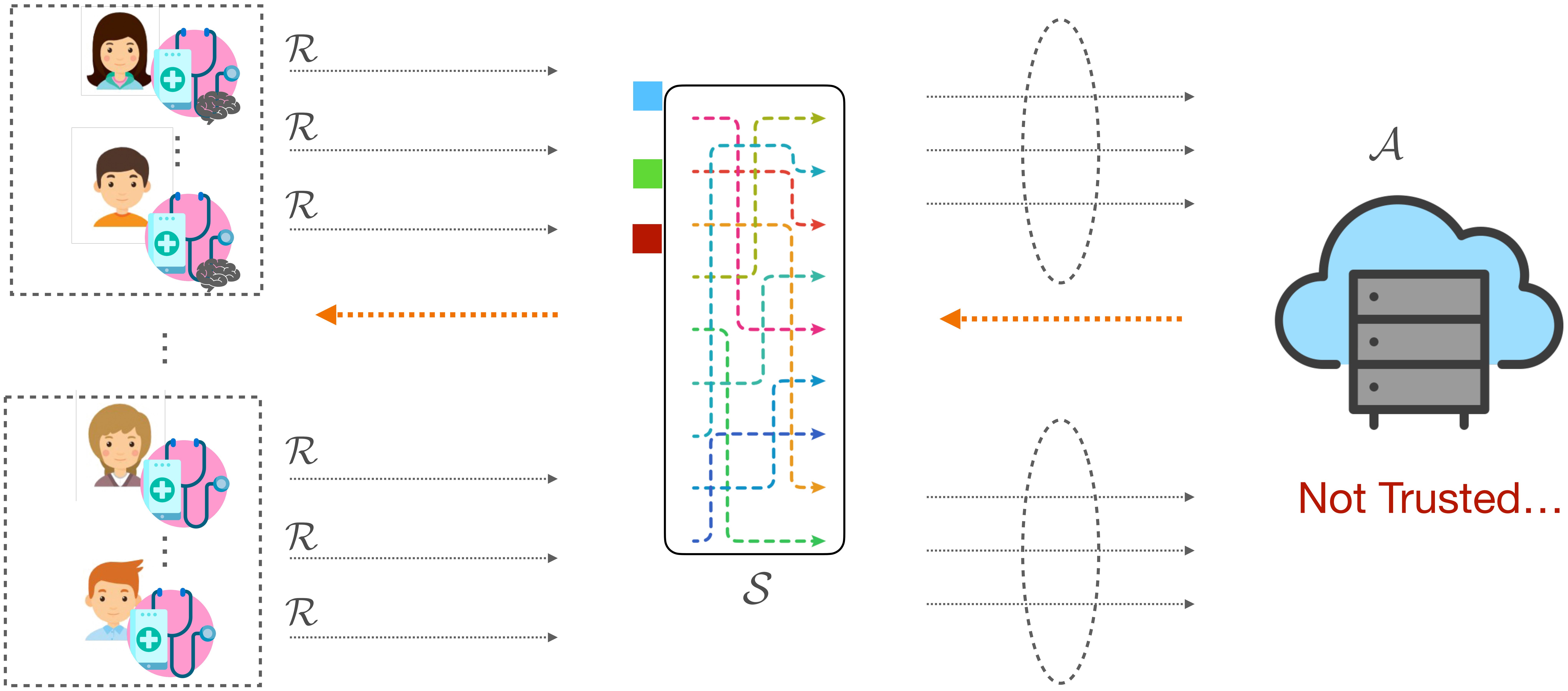
A Generic Private LinUCB

Illustration



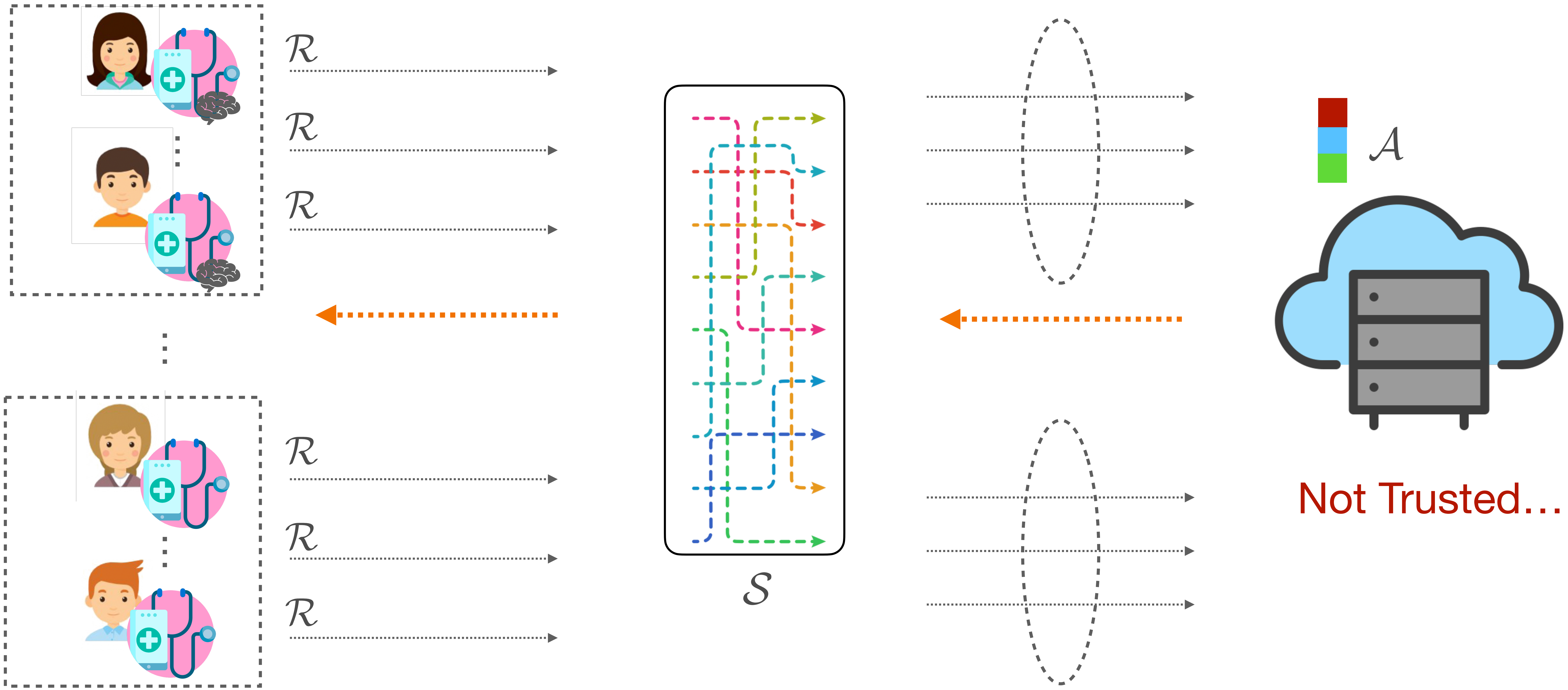
A Generic Private LinUCB

Illustration



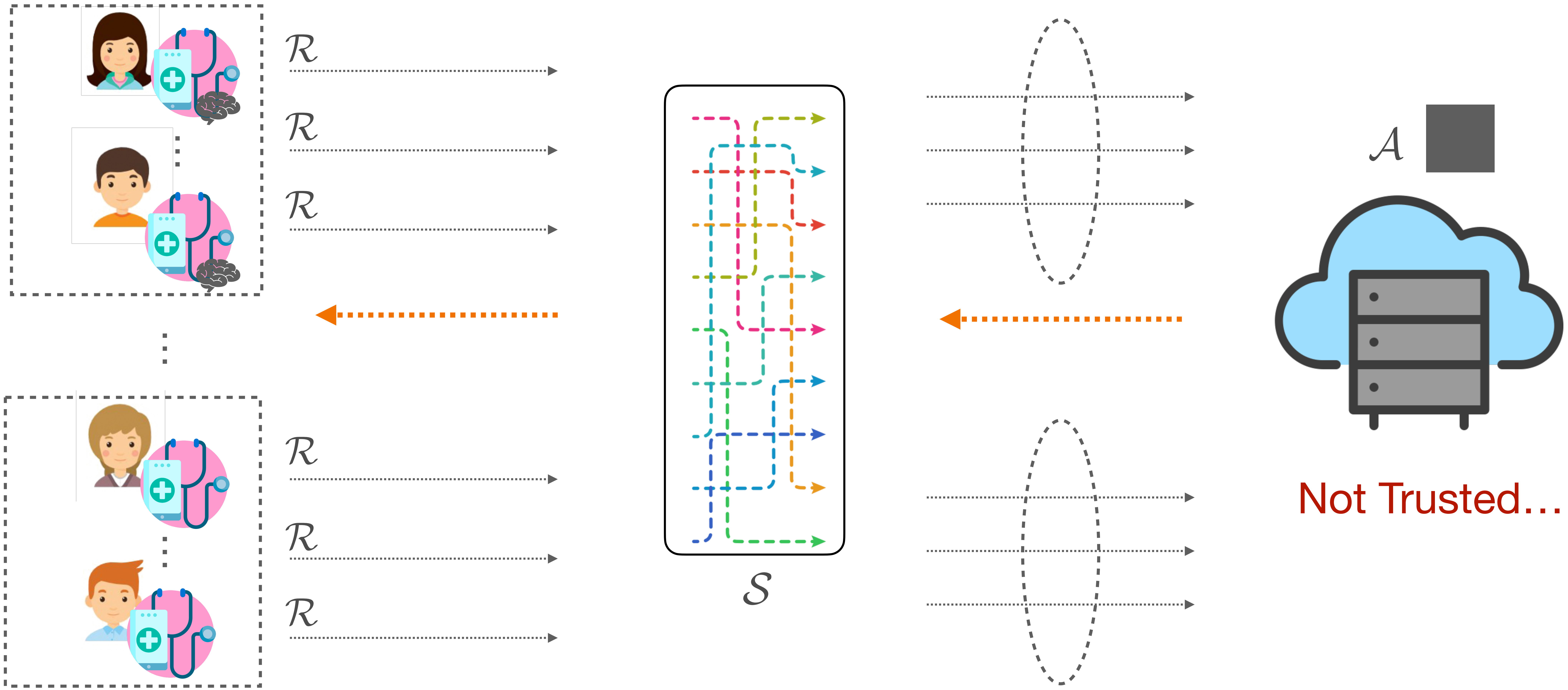
A Generic Private LinUCB

Illustration



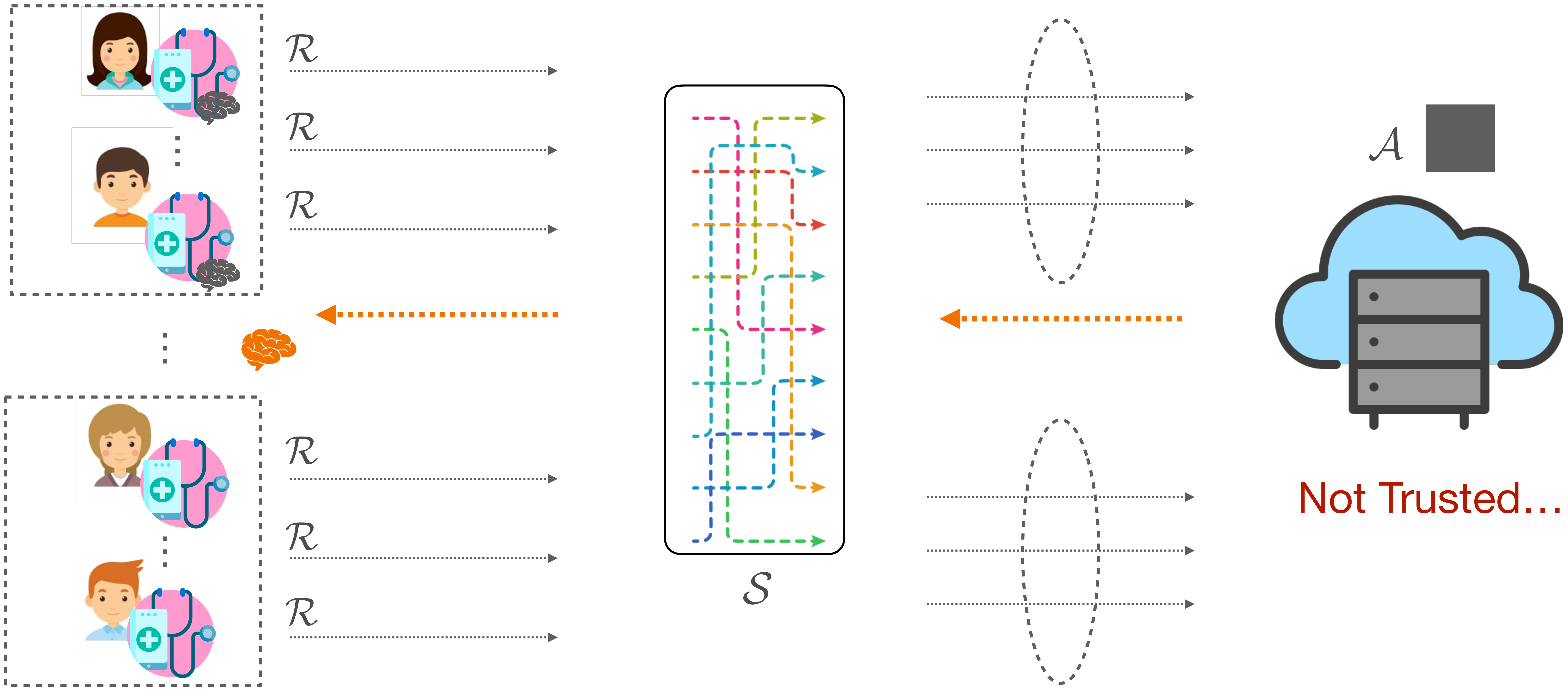
A Generic Private LinUCB

Illustration



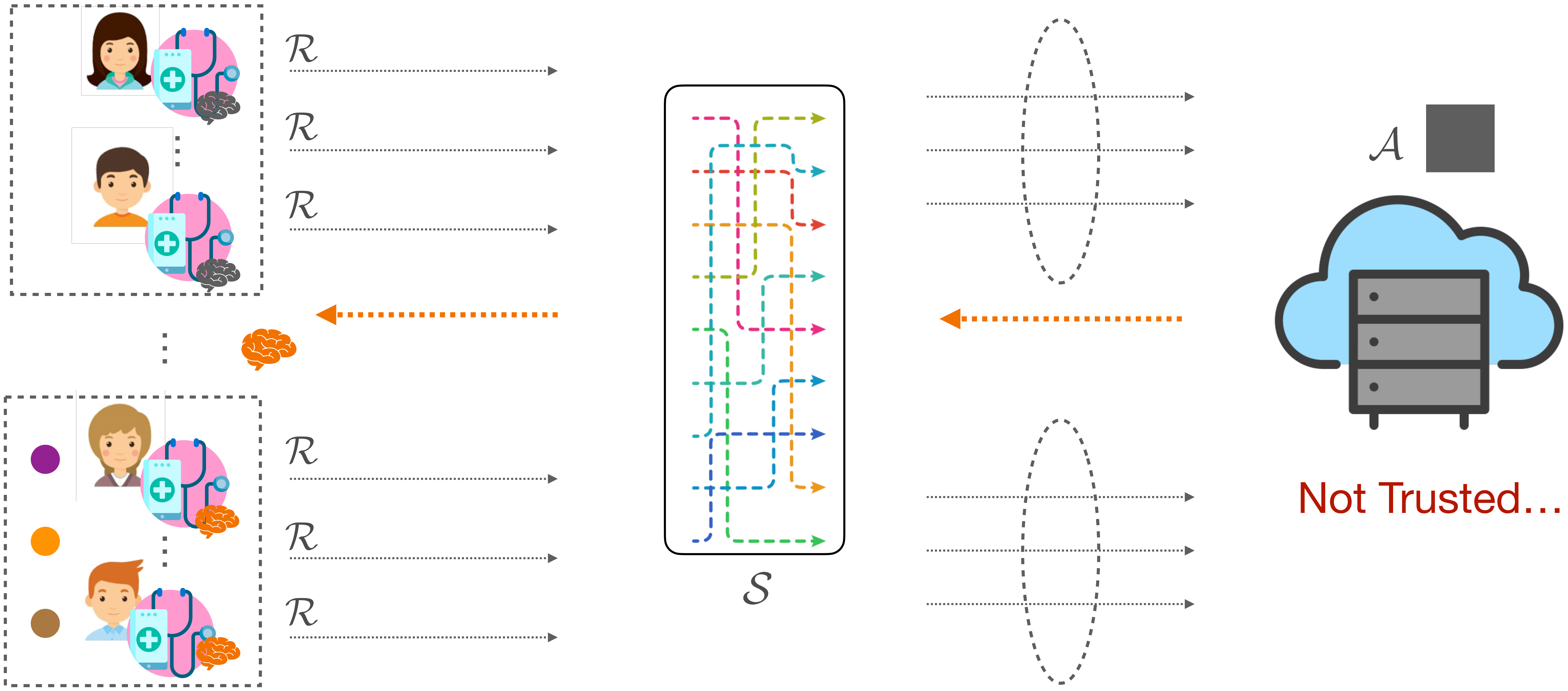
A Generic Private LinUCB

Illustration

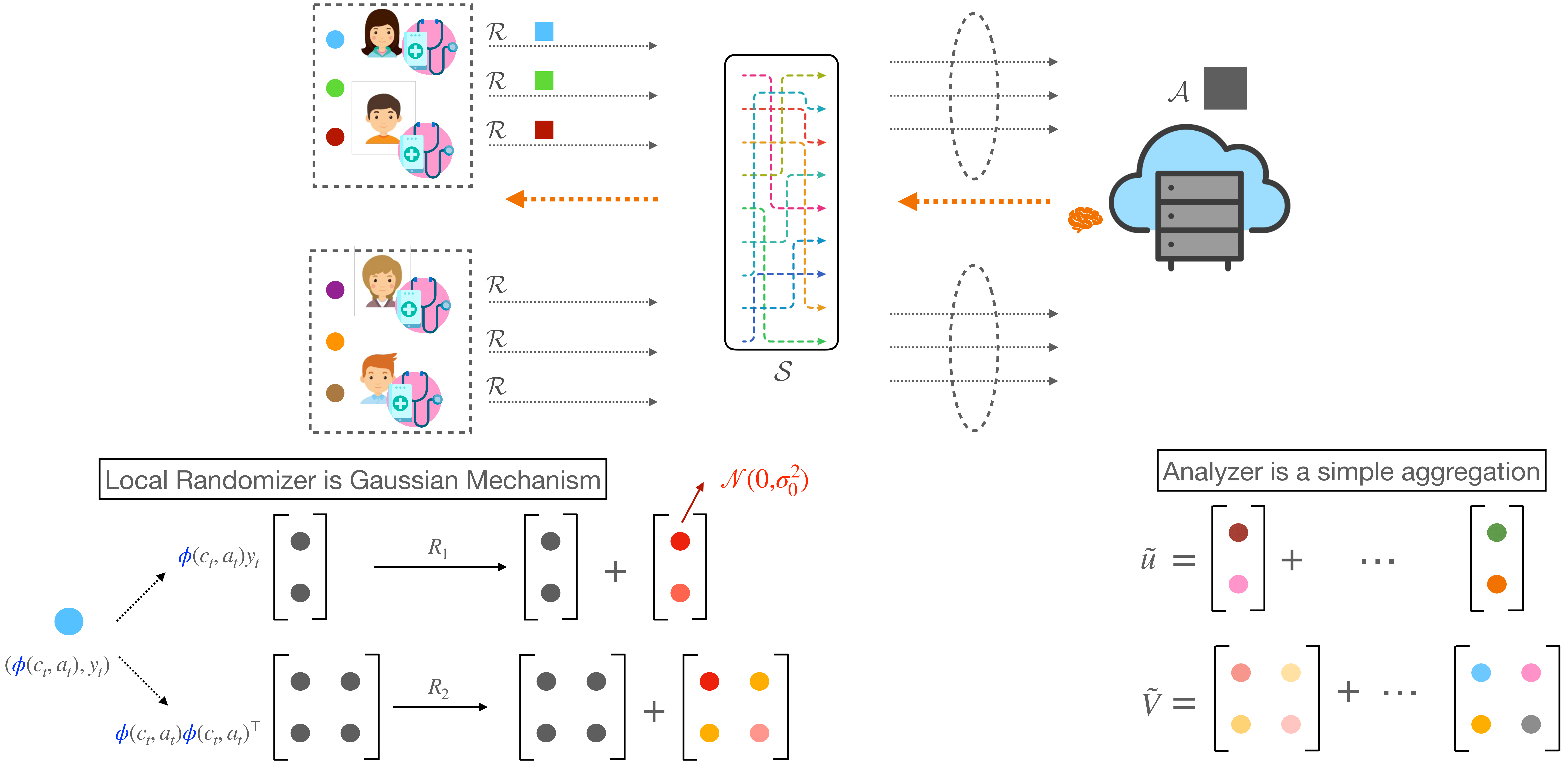


A Generic Private LinUCB

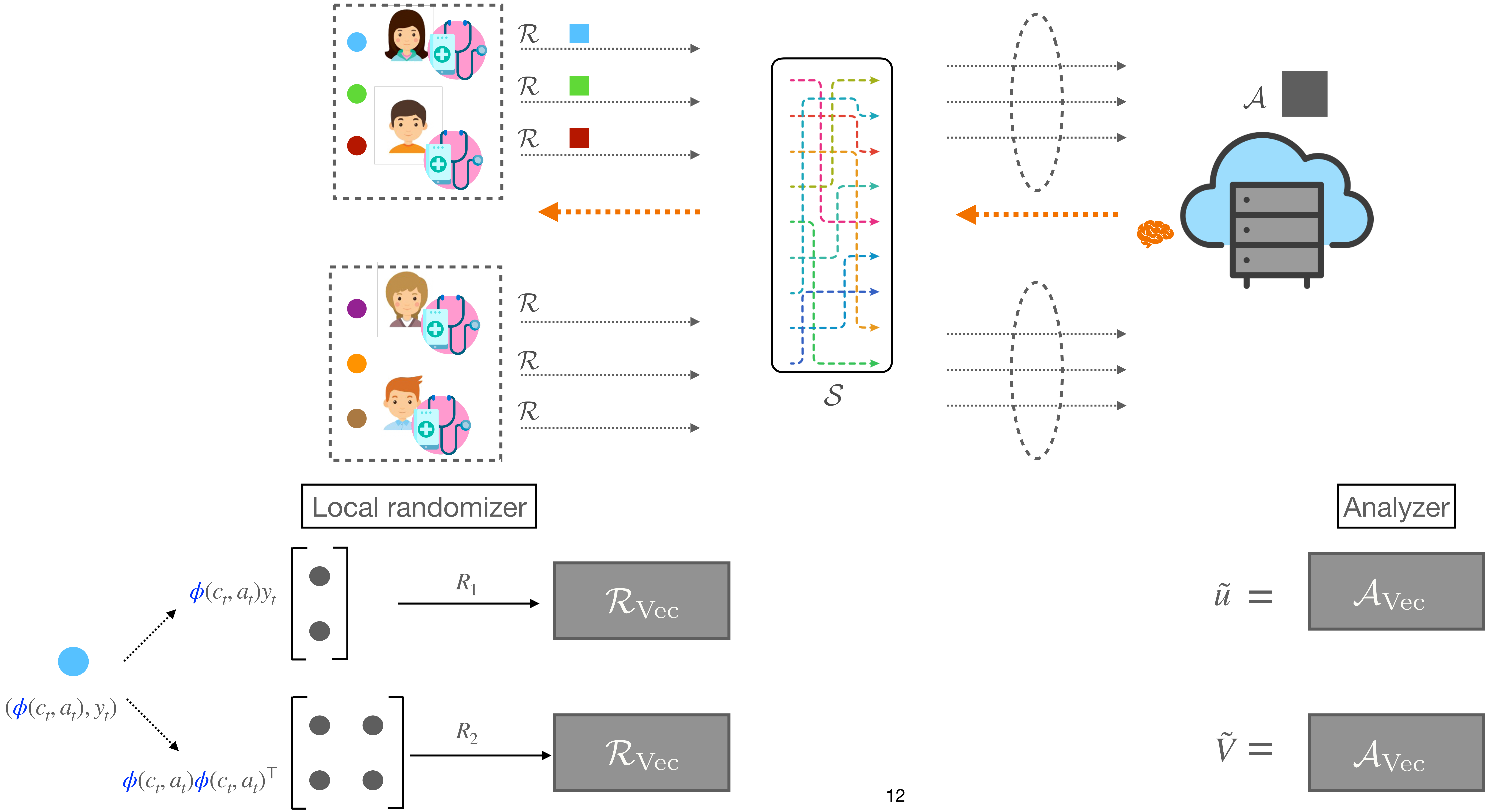
Illustration



P1: Amplification of Gaussian Mechanism



P2: Vector Sum for LCB



A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\mathrm{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T}d^{3/4}\right)$$

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\mathrm{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T}d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\text{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T}d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each user's noise is Gaussian with variance $\tilde{O}(1/(\epsilon^2 B))$ and a total of T such noise

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\text{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T} d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each user's noise is Gaussian with variance $\tilde{O}(1/(\epsilon^2 B))$ and a total of T such noise
- **SDP via Vector sum —** $\sigma^2 \approx O(T/(\epsilon^2 B))$

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\mathrm{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T} d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each user's noise is Gaussian with variance $\tilde{O}(1/(\epsilon^2 B))$ and a total of T such noise
- **SDP via Vector sum —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each batch is sub-Gaussian noise with variance $\tilde{O}(1/\epsilon^2)$ and a total of $M = T/B$ such noise

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\text{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T}d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each user's noise is Gaussian with variance $\tilde{O}(1/(\epsilon^2 B))$ and a total of T such noise
- **SDP via Vector sum —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each batch is sub-Gaussian noise with variance $\tilde{O}(1/\epsilon^2)$ and a total of $M = T/B$ such noise
- **Recover standard private bounds when** $B = 1$ **—** Central model: $\sigma^2 \approx \log T/\epsilon^2$ and Local model: $\sigma^2 \approx T/\epsilon^2$

A Generic Regret Bound

Applications

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\mathrm{Reg}(T) = \tilde{O}\left(dB + d\sqrt{T} + \sqrt{\sigma T} d^{3/4}\right)$$

- **SDP via LDP amplification —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each user's noise is Gaussian with variance $\tilde{O}(1/(\epsilon^2 B))$ and a total of T such noise
- **SDP via Vector sum —** $\sigma^2 \approx O(T/(\epsilon^2 B))$
  - Each batch is sub-Gaussian noise with variance $\tilde{O}(1/\epsilon^2)$ and a total of $M = T/B$ such noise
- **Recover standard private bounds when $B = 1$ —** Central model: $\sigma^2 \approx \log T/\epsilon^2$ and Local model: $\sigma^2 \approx T/\epsilon^2$
- **Batched central and local models … improve non-private batch LinUCB…**

Returning Users

Guarantees

Lemma

Let noise assumption hold. Our generic algorithm satisfies a high probability regret bound

$$\text{Reg}(T) = \tilde{O}\left(dT/M + d\sqrt{T} + \sqrt{\sigma T}d^{3/4}\right)$$

- **Shuffle model —** scale ϵ by $1/\sqrt{M}$ for (ϵ, δ)-SDP
  - As a result, total noise changes from $\sigma^2 \approx O(M/\epsilon^2)$ to $\sigma^2 \approx O(M^2/\epsilon^2)$
- **Central model —** scale ϵ by $1/M_0$ for (ϵ, δ)-DP in the central model
  - As a result, total noise changes from $\sigma^2 \approx O(\log T/\epsilon^2)$ to $\sigma^2 \approx O(M_0^2 \log T/\epsilon^2)$

If $M = M_0 = T^{1/3}$, both models have the same regret $\tilde{O}(T^{2/3})$!

*

Open Problems

Open Problems

- Can we close the gap?

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?
- **Can we achieve pure DP in all three models?**

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?
- **Can we achieve pure DP in all three models?**
  - The key challenge is a non-trivial matrix concentration bound with sub-exponential tails

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?
- **Can we achieve pure DP in all three models?**
  - The key challenge is a non-trivial matrix concentration bound with sub-exponential tails
  - For shuffle model, additional care is required, see our recent work [CZ'22 Distributed DP in MAB]

Open Problems

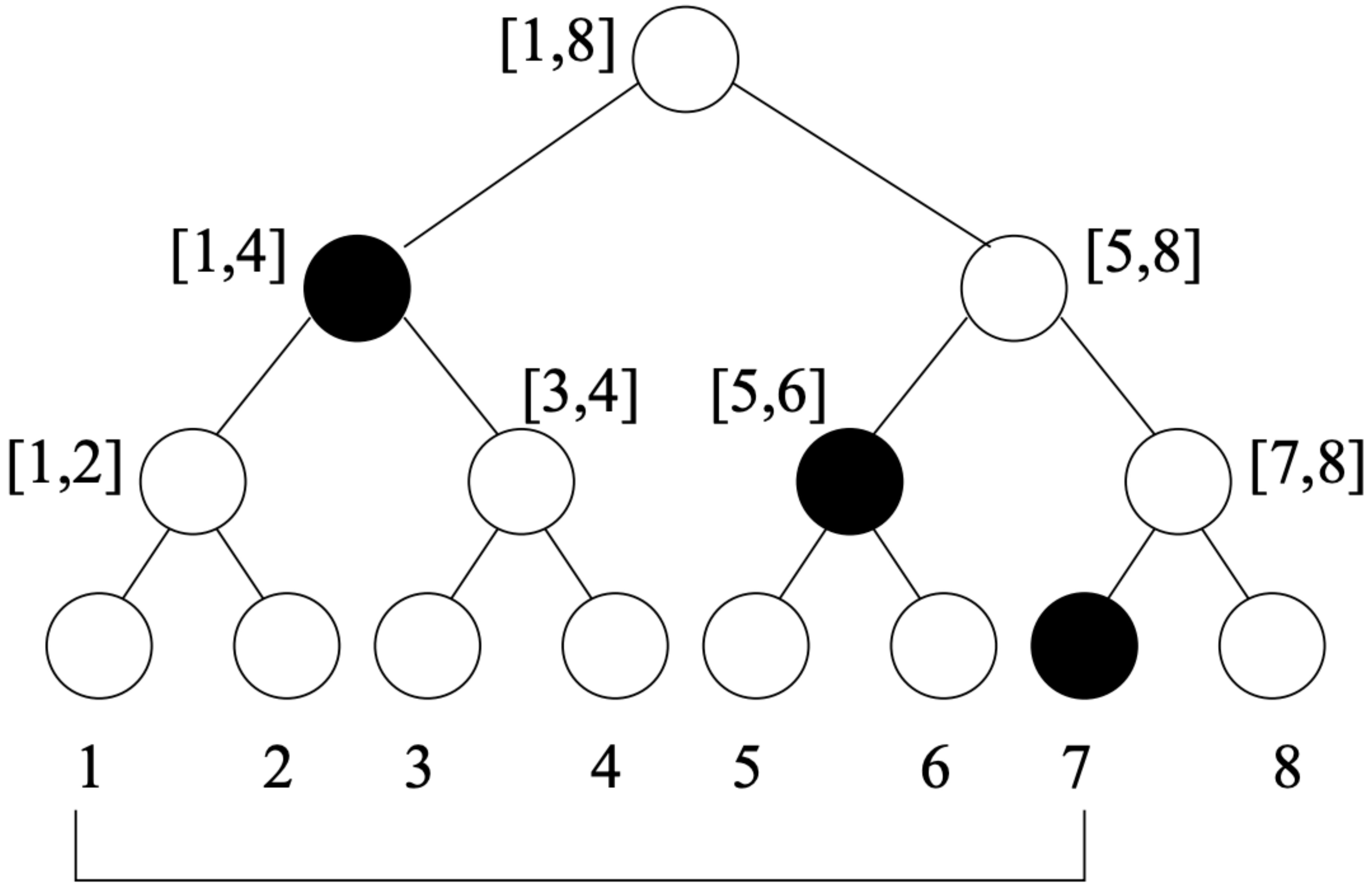
- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?
- **Can we achieve pure DP in all three models?**
  - The key challenge is a non-trivial matrix concentration bound with sub-exponential tails
  - For shuffle model, additional care is required, see our recent work [CZ'22 Distributed DP in MAB]
- **Can we do adaptive batch schedule (i.e., rarely-switching) in private case?**

Open Problems

- **Can we close the gap?**
  - What's the lower bound for local model? i.e., Can $O(T^{3/4})$ be improved?
  - Or, can one further improve $O(T^{3/5})$ in the shuffle model?
- **Can we achieve pure DP in all three models?**
  - The key challenge is a non-trivial matrix concentration bound with sub-exponential tails
  - For shuffle model, additional care is required, see our recent work [CZ'22 Distributed DP in MAB]
- **Can we do adaptive batch schedule (i.e., rarely-switching) in private case?**
  - The key challenge is that standard determinant trick fails, ($V_t \succeq V_{\tau_t}$, where $\tau_t < t$ is the recent update time)

Thank you!

Backup



(b) The sum of time steps 1 through 7 can be obtained by adding the **p-sums** corresponding to the black nodes.