Decentralized Learning (as an enabler) for Decentralized Online Services

Sonia BEN MOKHTAR
Institut Data IA

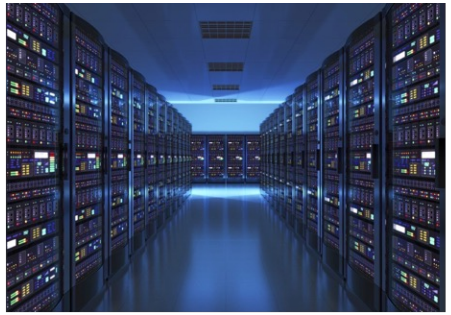
10/01/2023

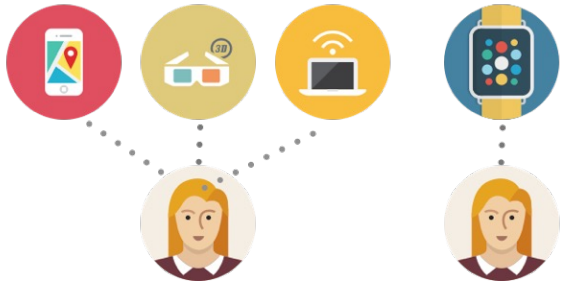
Who am I?

- Head of the DRIM team @LIRIS lab
  - Distrubuted systems
    - Dependability
    - Privacy (e.g., location privacy, private web search, private recommender systems)
    - Performance
  - Information Retrieval

  - Increasing interest for Distributed Learning
    - Numerous challenges in terms of dependability, privacy & performance

Today’s Online Services





An example: Web search

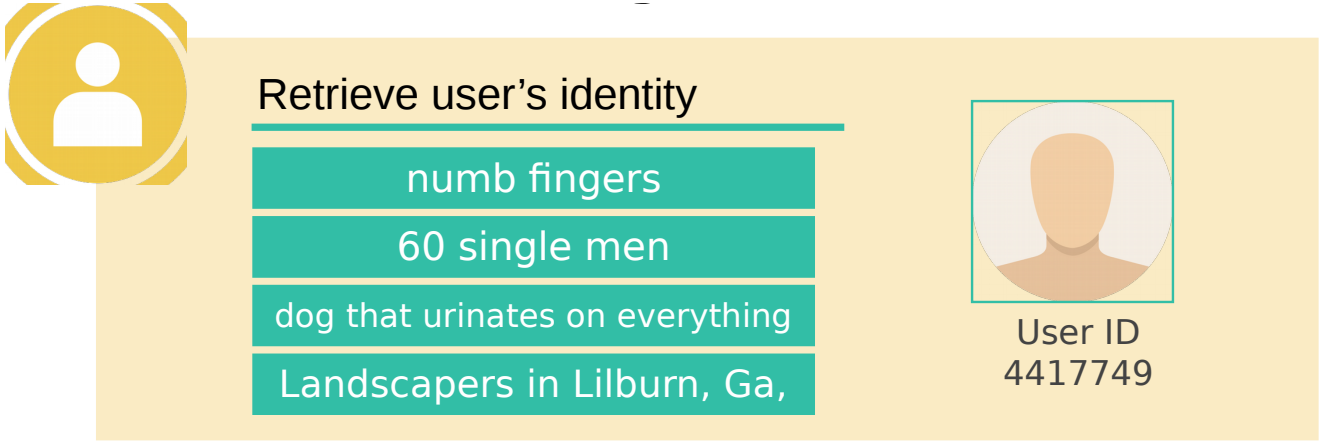
Every day, millions of users are querying **SEARCH ENGINES**

> We also use this information [*that we collect from all of our services*] to offer you tailored content – like giving you more **relevant search results** and **ads**.

http://www.google.com/policies/privacy/

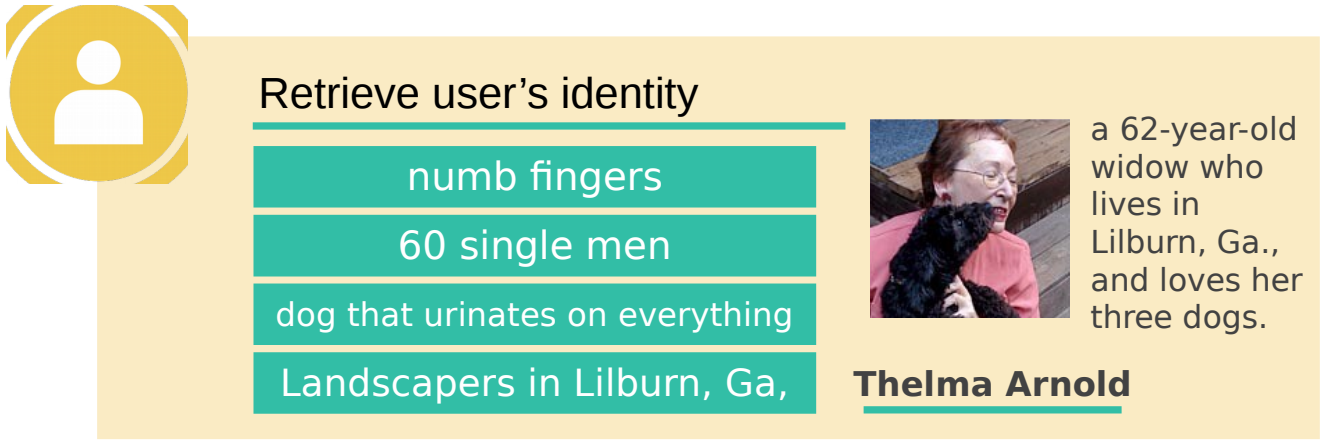
USER PROFILES

Web Search: Privacy Threats



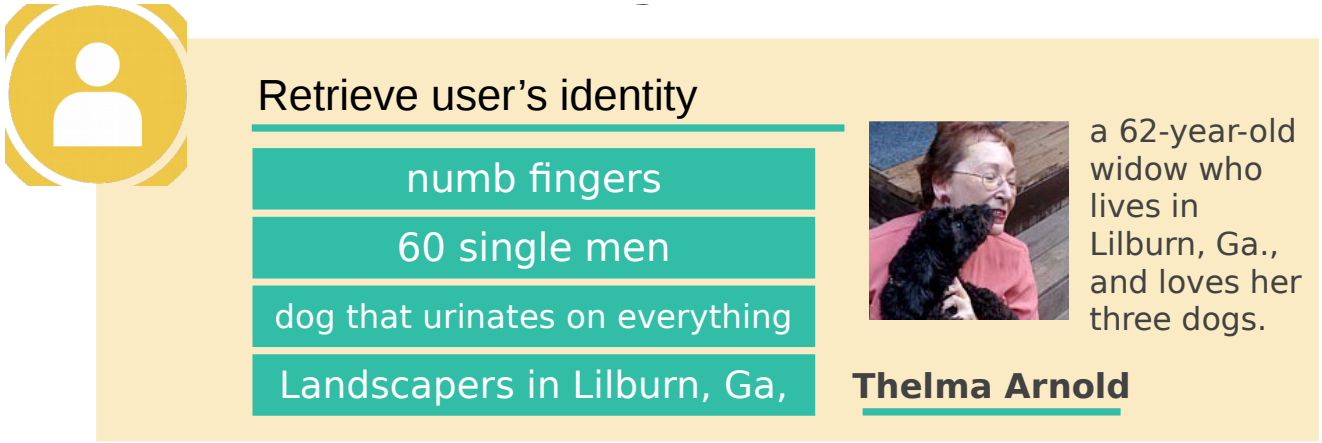
Barbaro, Michael, Tom Zeller, and Saul Hansell. "A face is exposed for AOL searcher no. 4417749." New York Times 9.2008 (2006): 8For.

Web Search: Privacy Threats

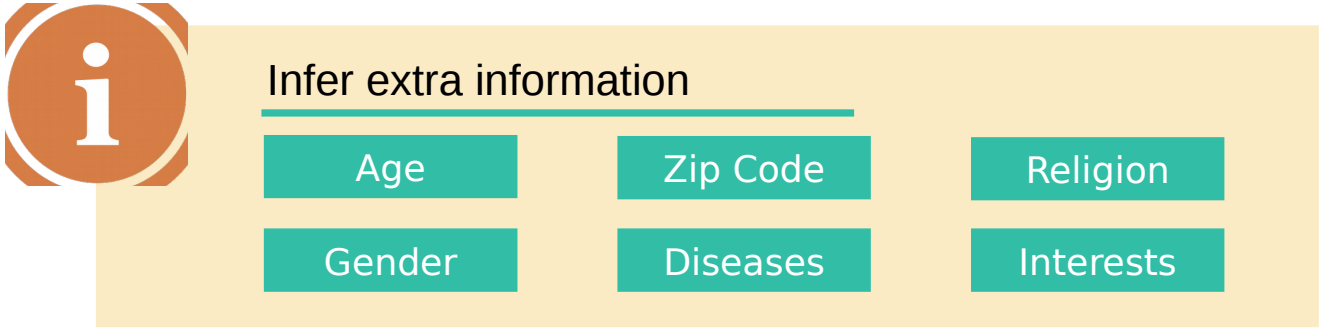


Barbaro, Michael, Tom Zeller, and Saul Hansell. "A face is exposed for AOL searcher no. 4417749." New York Times 9.2008 (2006): 8For.

Web Search: Privacy Threats



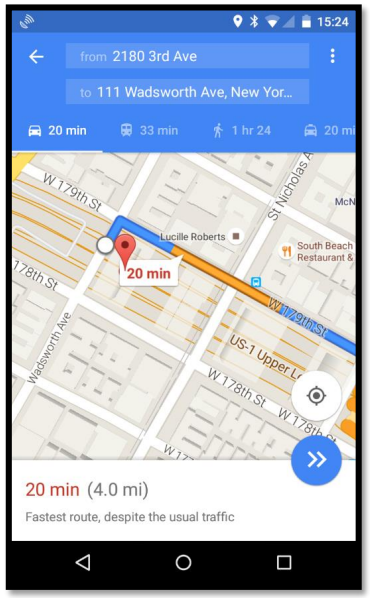
Barbaro, Michael, Tom Zeller, and Saul Hansell. "A face is exposed for AOL searcher no. 4417749." New York Times 9.2008 (2006): 8For.



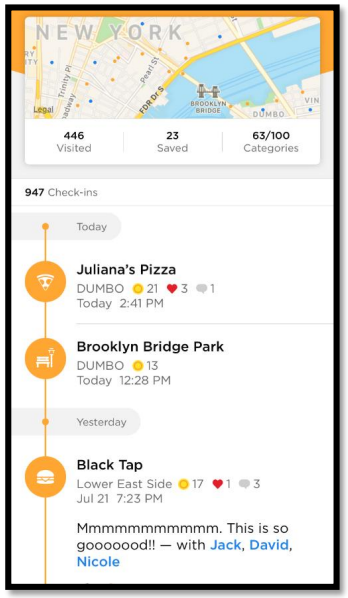
Jones, Rosie, et al. "I know what you did last summer: query logs and user privacy." Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. ACM, 2007.

Another example: Location-based services

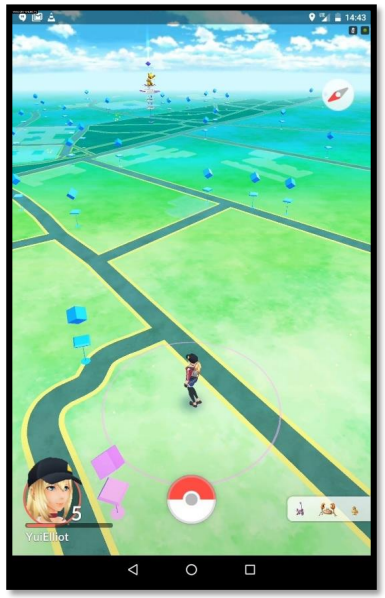
Location based services are very useful for users



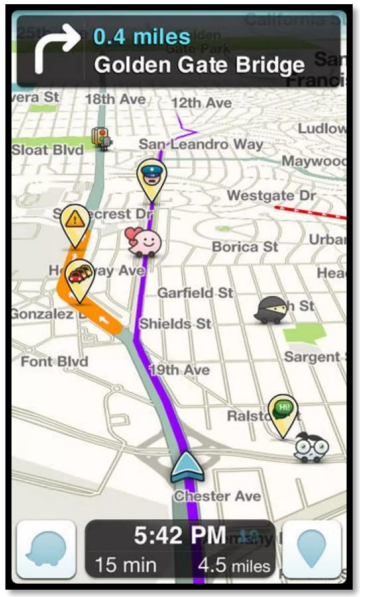
Navigation
e.g., Google Maps



Social Network
e.g., Foursequare



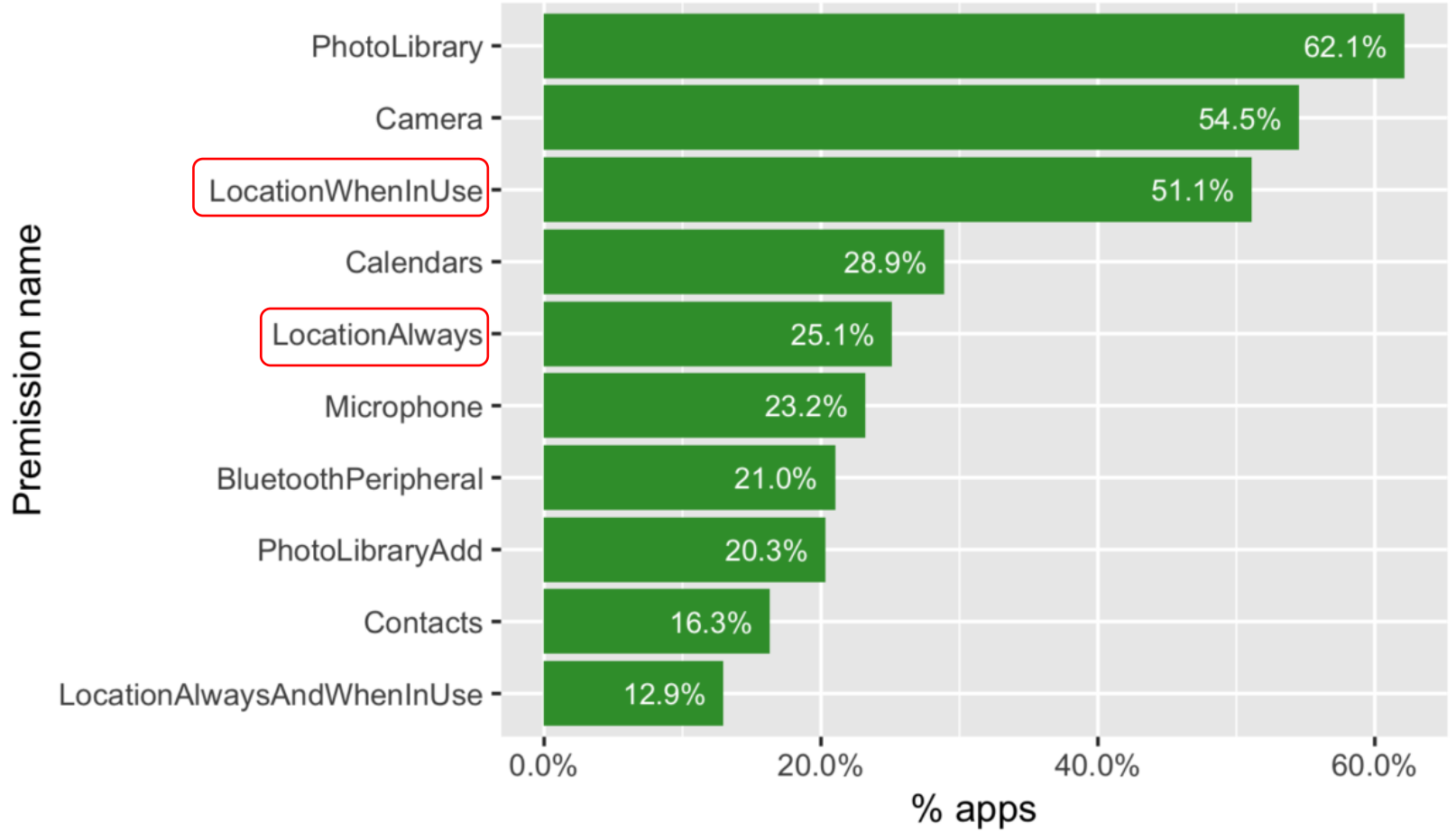
Video Games
e.g., Pokemon Go



Crowd-Sensing
e.g., Waze

Location data collection

Most asked permissions of 30.000 sampled apps in the Apple Store



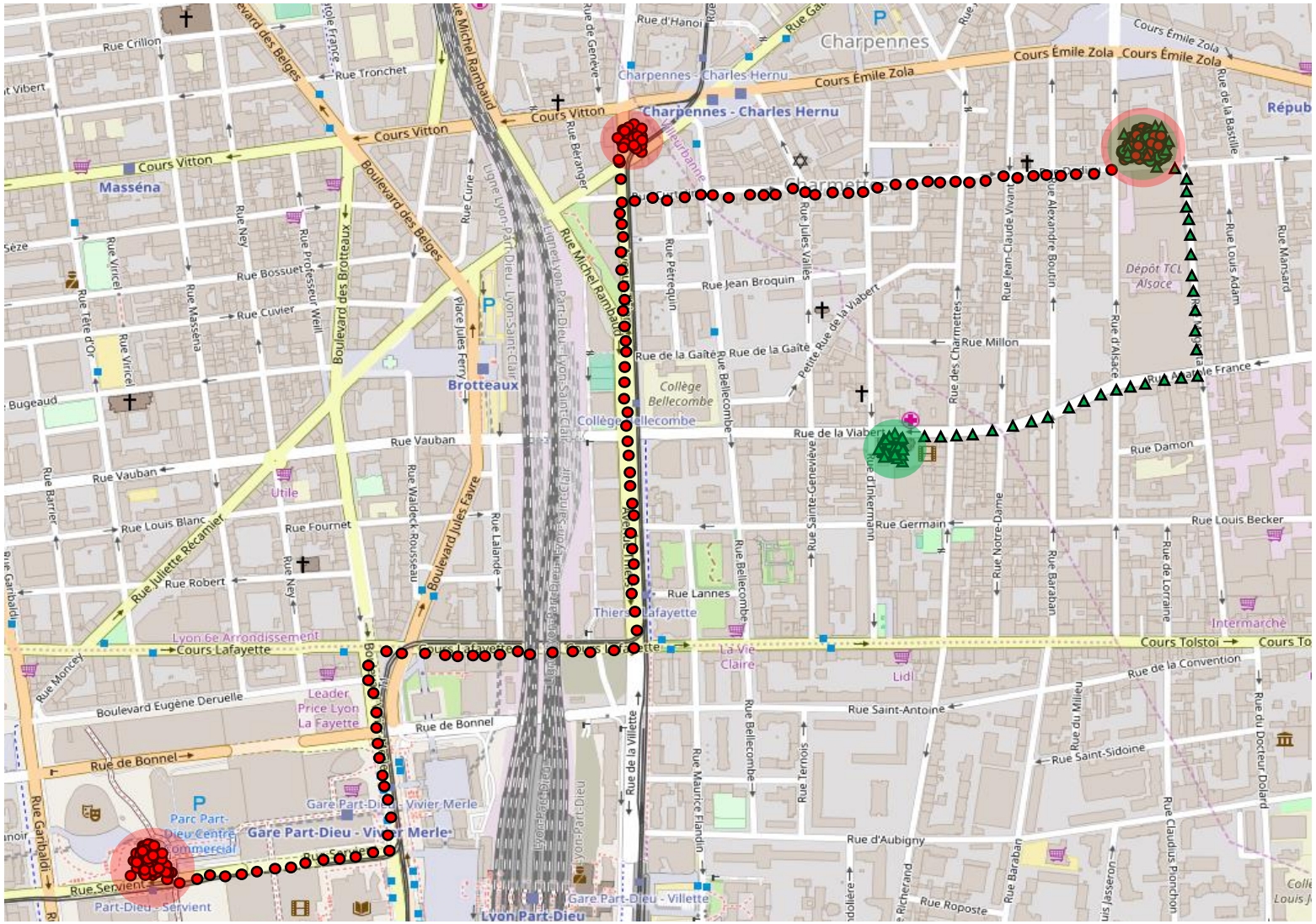
Location-based Services: Threats

Point of Interest (POI) of A

Point of Interest (POI) of B

Record of User A

Record of User B



Location-based Services: Threats

Home

Workplace

Medical Facility

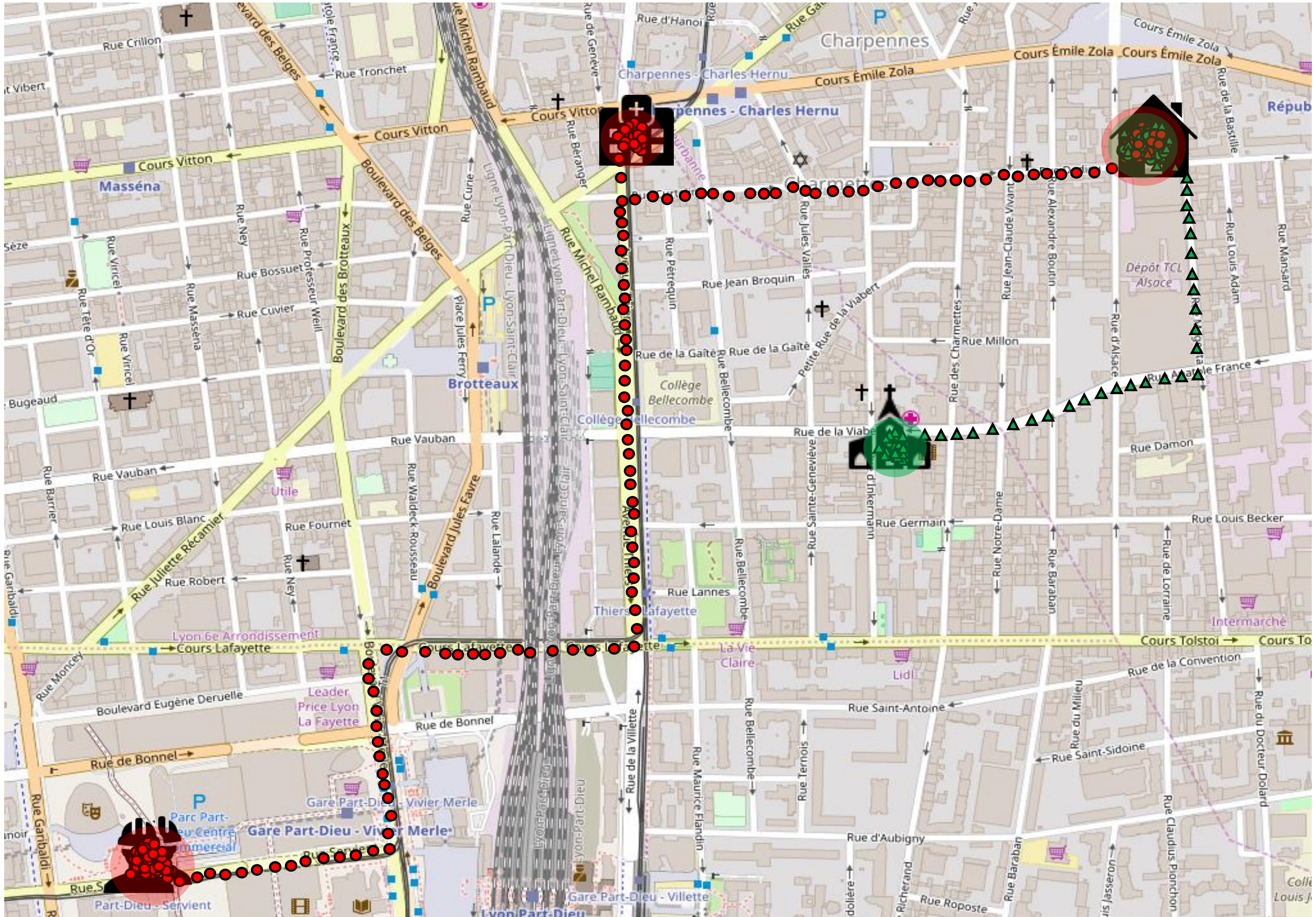
Place of Worship

Point of Interest (POI) of A

Point of Interest (POI) of B

Record of User A

Record of User B



Threats Illustrated

Angry Birds and 'leaky' phone apps targeted by NSA and GCHQ for user data

- US and UK spy agencies piggyback on commercial data
- Details can include age, location and sexual orientation
- Documents also reveal targeted tools against individual phones

QUARTZ

EXCLUSIVE

Google collects Android users' locations even when location services are disabled

By Keith Collins · November 21, 2017

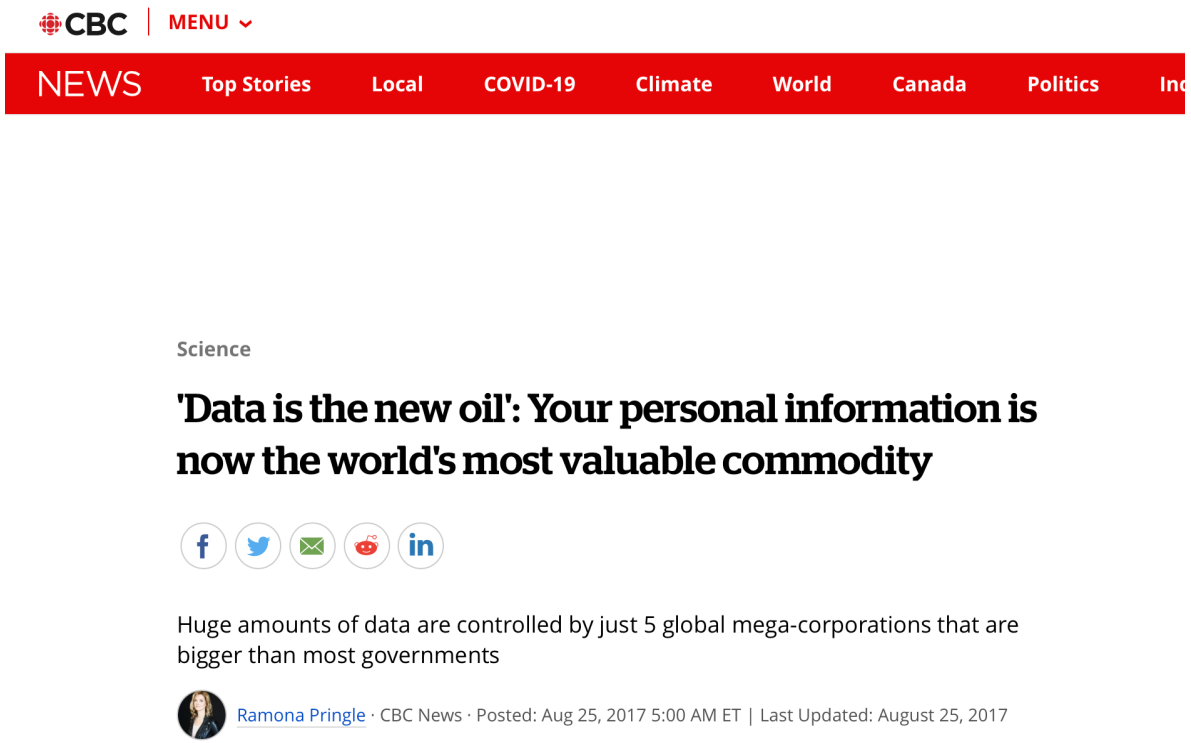
QUARTZ

SWIPED

Dating app Tinder briefly exposed the physical location of its users

By Zachary M. Seward in California · July 23, 2013

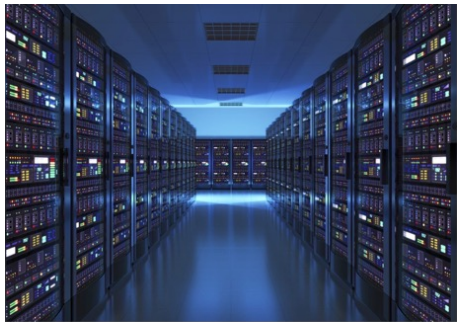
Data is the new oil

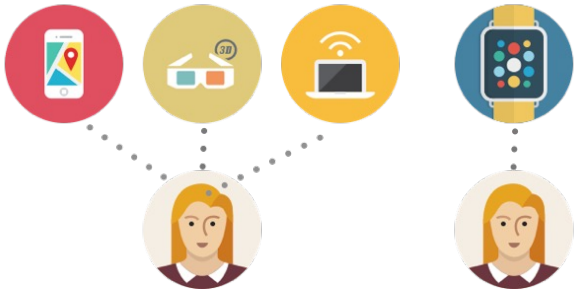


- *"..the corporate giants are collecting information about every aspect of our lives, our behaviour and our decision-making…"*
- Data is used by online services for
  - Improving their algorithms
  - Mastering strengths and vulnerabilities of suppliers, competitors and customers
  - Earning money through the Ad system ("*Google and Facebook control 88 per cent of all new internet advertising"*)

Today’s Online Services

- Heavily centralized (governance)
- Data-centric
- Open numerous threats





- Increased user awareness on privacy
- Legislator
  - GDPR, …


sif008.jpg
DANS LE FUTUR, ON PEUT RÊVER D'UN WEB DÉCENTRALISÉ...
...ET D'UN VOTE RANDOMISÉ...
TANT QU'ON Y EST !!

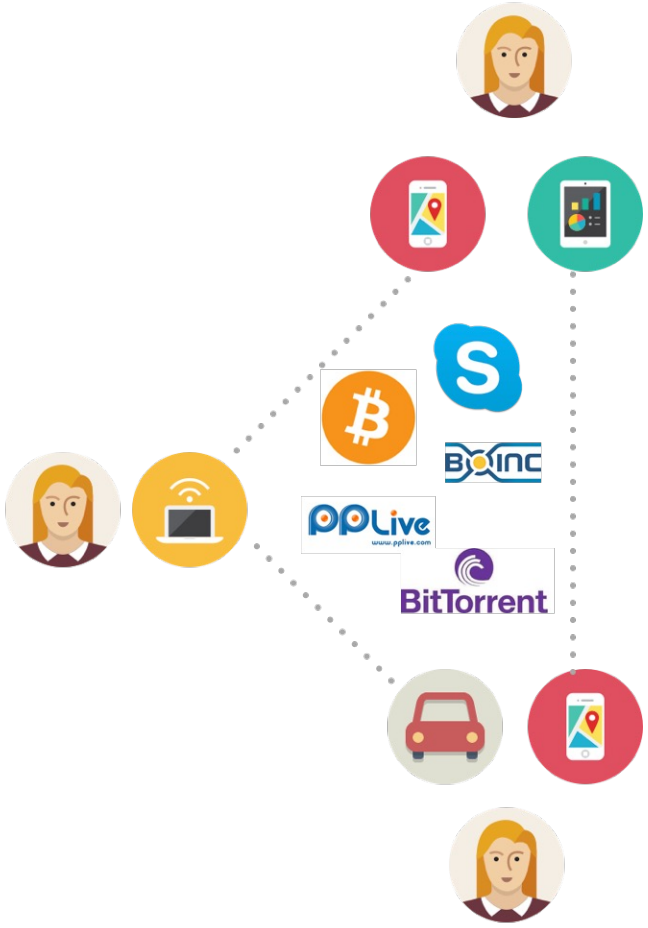
Decentralized Systems

- ”*a subset of distributed systems where multiple authorities control different components and no authority is fully trusted by all*” [Troncoso et al. PETS’17]

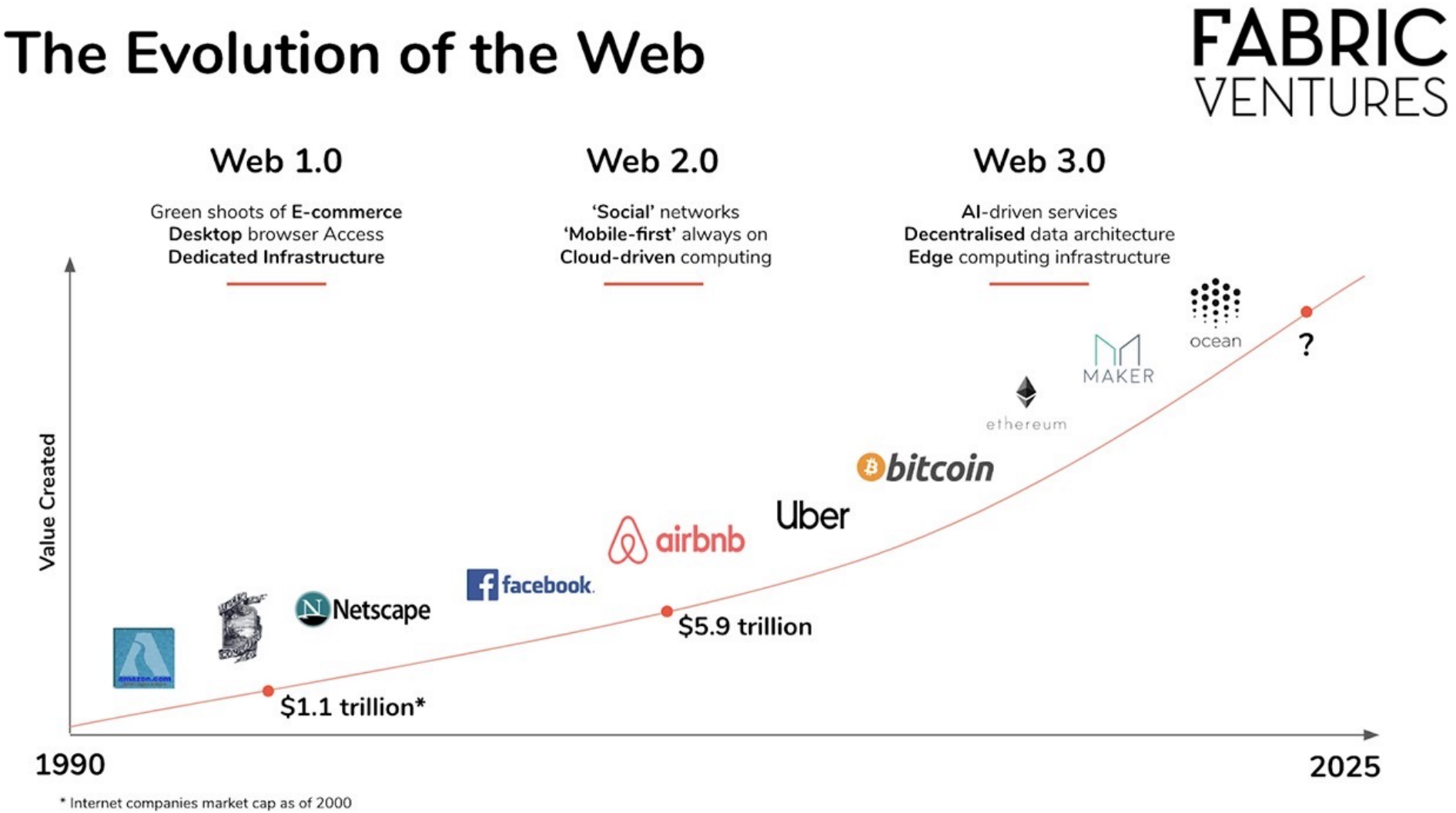
- Decentralization facets
  - Scalability/Openness
  - Resilience
  - Incentives

Decentralized Systems: not a new concept

- Peer-to-Peer systems (as opposed to client-server architectures)
- 1999: Napster file sharing system
  - Followed: Gnutella, G2, eDonkey, BitTorrent, PPlive, ToR…
- Tim Berners-Lee's vision for the World Wide Web was close to a P2P*“: each user of the web would be an active editor and contributor, creating and linking content to form an interlinked "web" of links”.*



Web 3.0: a new wave of Web Decentralization



There will be no *decentralized services* without *decentralized learning*

Online Services Heavily Rely on ML algorithms

- "*Facebook/Meta would collapse if you remove ML algorithms*" said Y. Lecun.

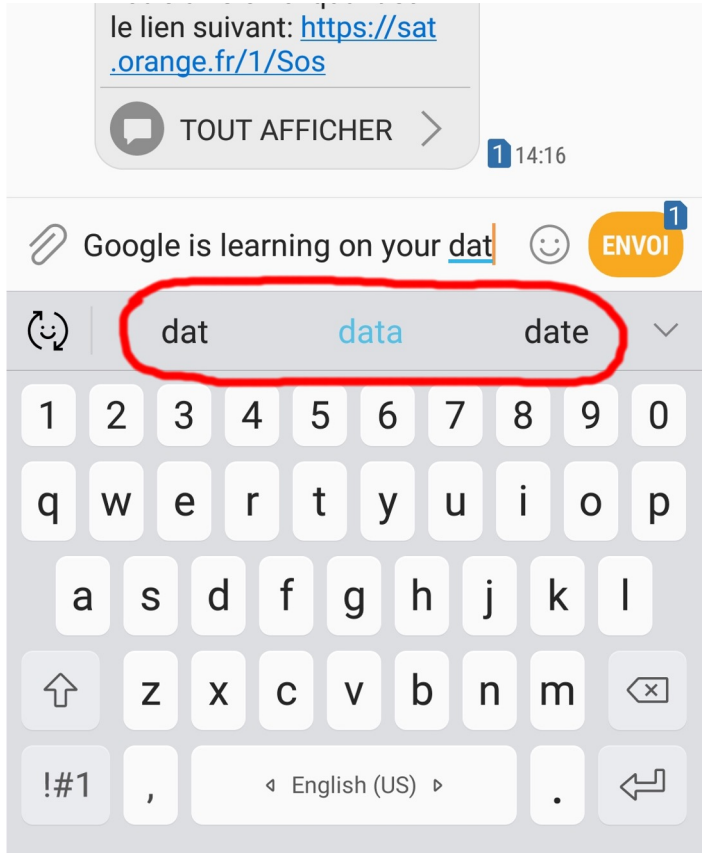
Models	Services
Support Vector Machines (SVM)	Facer (User Matching)
Gradient Boosted Decision Trees (GBDT)	Sigma
Multi-Layer Perceptron (MLP)	Ads, News Feed, Search, Sigma
Convolutional Neural Networks (CNN)	Lumos, Facer (Feature Extraction)
Recurrent Neural Networks (RNN)	Text Understanding, Translation, Speech Recognition

TABLE I
MACHINE LEARNING ALGORITHMS LEVERAGED BY PRODUCT/SERVICE.

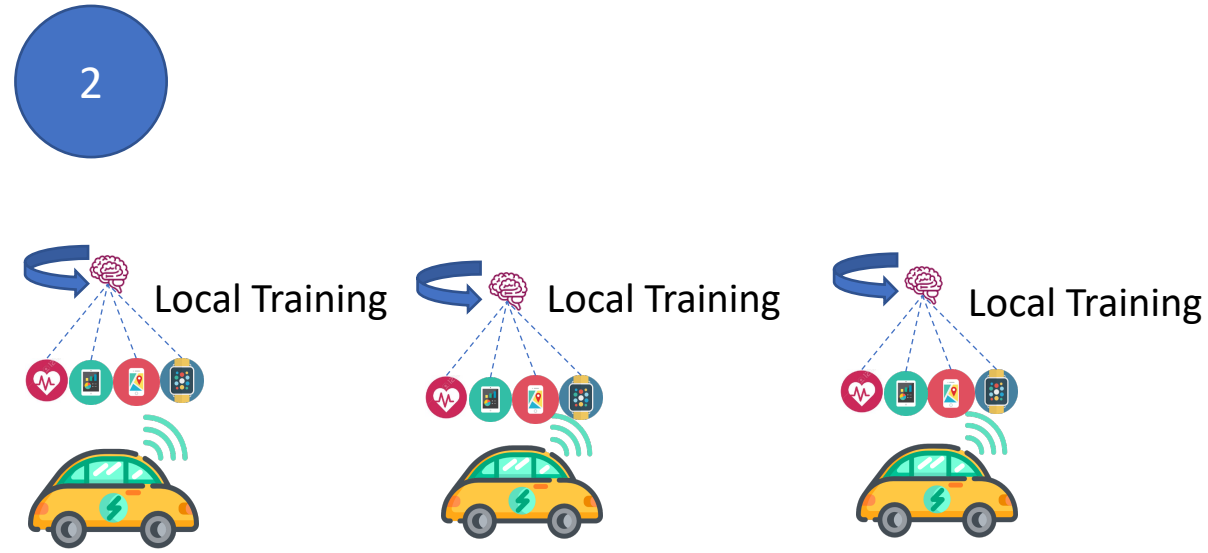
Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective. Facebook Inc. HPCA'18.

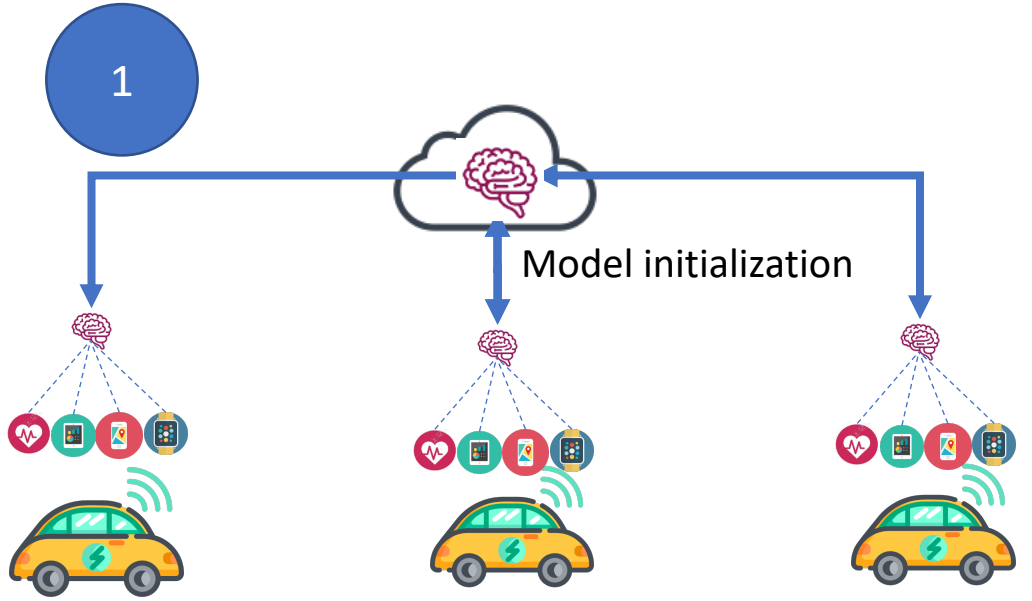
Federated Learning : a Natural Candidate

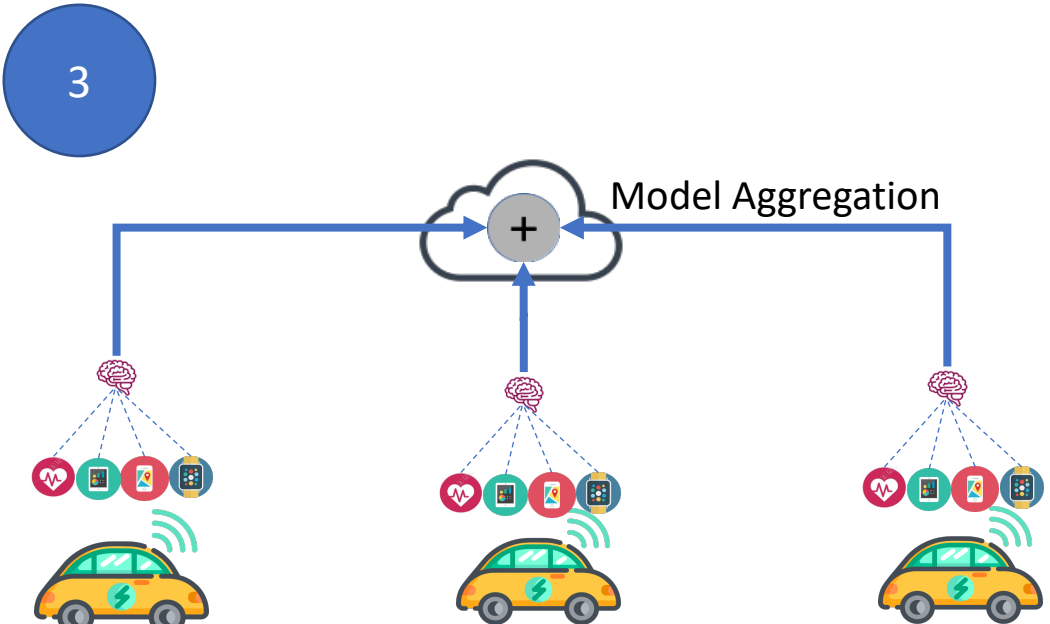
- Federated learning (FL) aims at collaboratively train ML models while keeping the data decentralized
- 2016: Initially proposed by Google Research for training the Gboard (Google Android Keyboard)
- 2022: thousands of research papers published every year
- Interest coming from varius communities
  - AI/ML, optimization, distributed systems, networks, security, privacy, dependability, …
- Some real world deployments
- Libraries: PySyft, TensorFlow Federated, FATE, Flower, Substra...

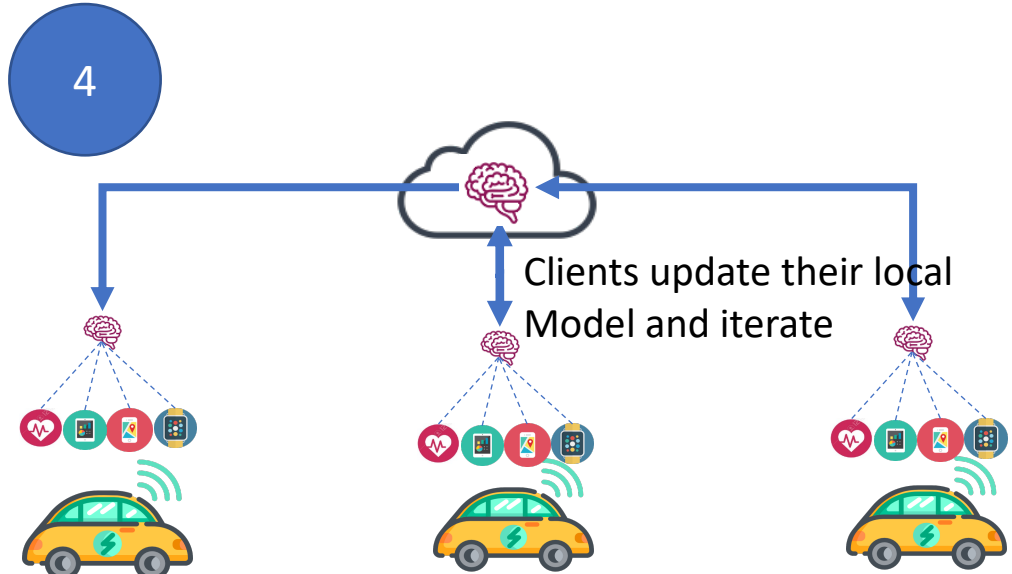


Federated Learning









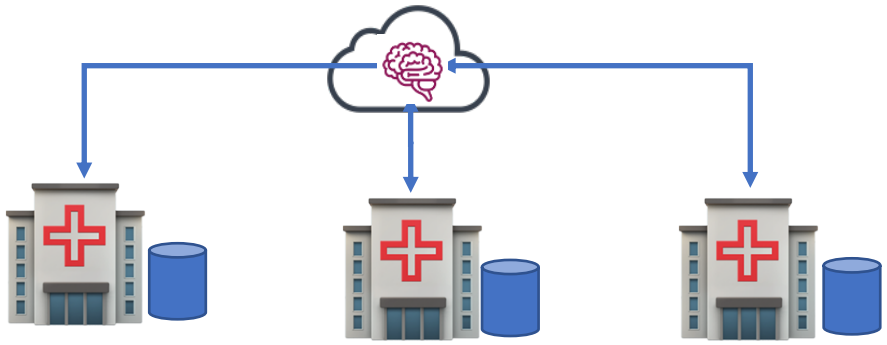
FL Key Characteristics

- Data is generated locally
- Data is imbalanced and not independent and identically distributed (non-i.i.d)
- Privacy/Robustness issues
  - Model updates may embedd knowledge about the participants
  - Limited reliability/availability of participants
  - Robustness against selfish parties
  - Robustness against malicious parties
  - ….

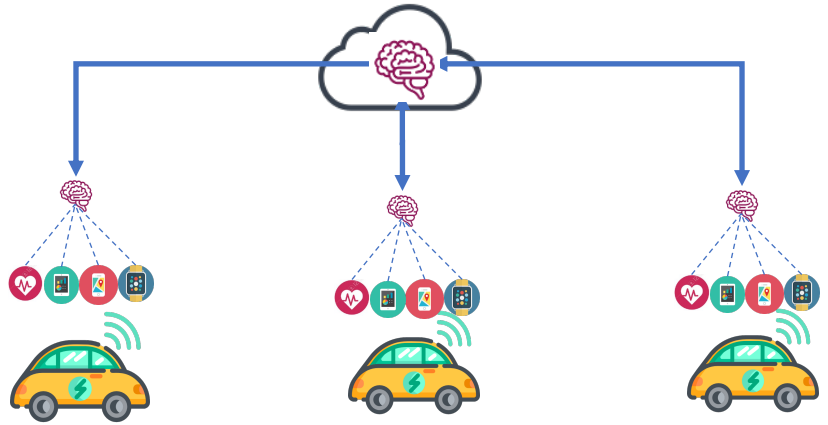
Slide by. A. Bellet

Cross-silo vs Cross-device FL

- Cross-silo
  - ~2-100 parties
  - Medium/large dataset per party
  - Reliable/available parties
  - Parties are trusted

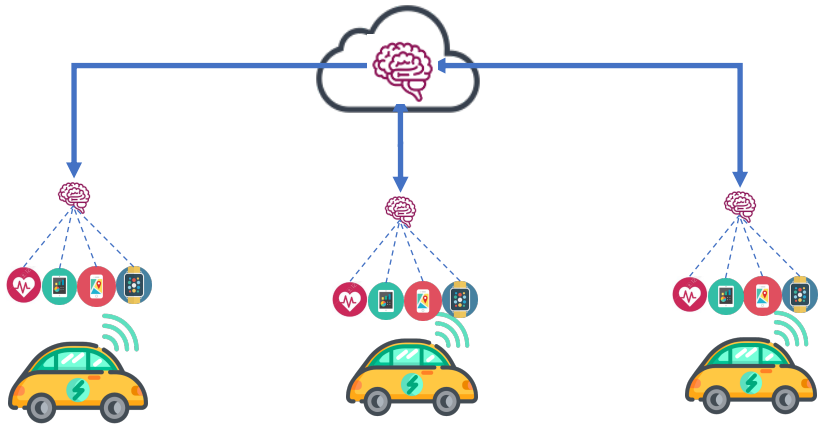


- Cross-device
  - Massive number of parties (millions)
  - Small dataset per party
  - Limited reliability/availability
  - Some parties may be malicious

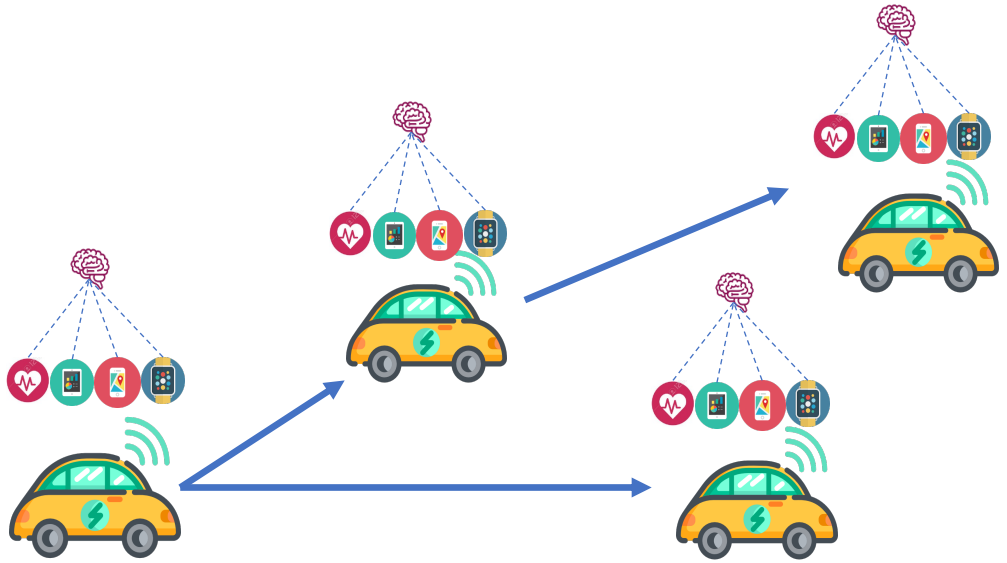


Server Orchestrated vs. Fully Decentralized

- Orchestrated
  - Server-client communication
  - Global coordination, global aggregation
  - Server is a single point of failure and may become a bottleneck



- Decentralized
  - Device to device communication
  - No global coordination, local aggregation
  - Naturally scales to a large number of devices



Decentralizing Online Services with Distributed/Decentralized Learning

- Usecases
  - **Decentralizing Recommender Systems with Gossip Learning**
    - PhD Yacine Bellal [Ubicomp'22]
  - FL-based Location Privacy
    - PhD Besma Khalfoun [Ubicomp'21][Middleware'20]
  - Decentralized and Secure Web Search with Trusted Execution Environments
    - PhD Matthieu Bettinger + Aghiles Ait Messaoud [Middleware'22]

- Addressed challenges
  - **Personalization**
  - Privacy
  - Robustness

Decentralized Recommender Systems over Gossip Learning

Joint work with Yacine Belal, Vlad Nitu & Aurélien Belet

Presented@Ubicomp22

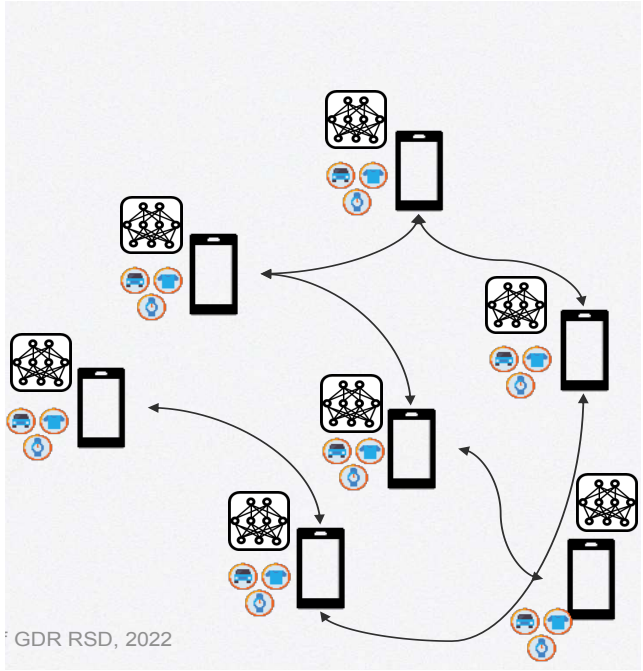
- Recommender Systems are everywhere.
- Netflix values recommendations at half a billion dollars to the company.
- LinkedIn job matching algorithms improves performance by 50%

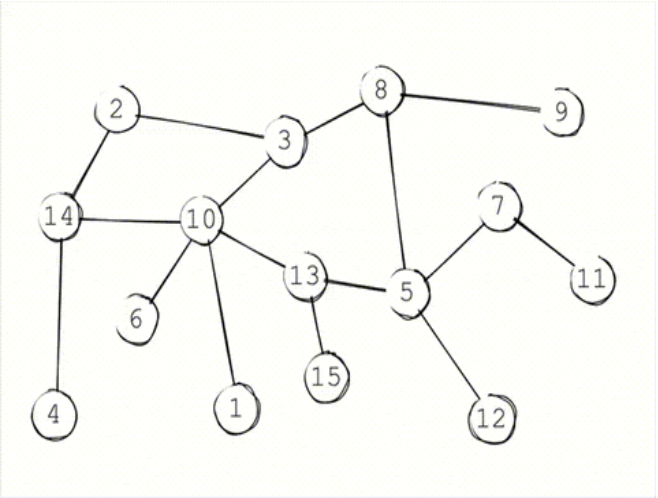
Recommender Systems (RecSys)

- Users rate Items
- RecSys predicts items a user might like
- Centralized
  - User-based/Item-based collaborative filtering
    - Recommends to a given user items that similar users have liked
  - Matrix Factorization (MF)
  - Neural Collaborative Filtering (DNN inspired from MF)
- Federated
  - Generalized Matrix Factorization (GMF)

Gossip Learning

- Each node owns local data and maintains a local model
- Nodes exchanges their model updates asynchronously
- Each node aggregates the received models (it acts as *a server)*
- The objective could be
  - To train a model at each node that performs well wrt a local distribution (personalization) -> RecSys
  - To train a model at each node that performs well wrt a global distribution (generalization)



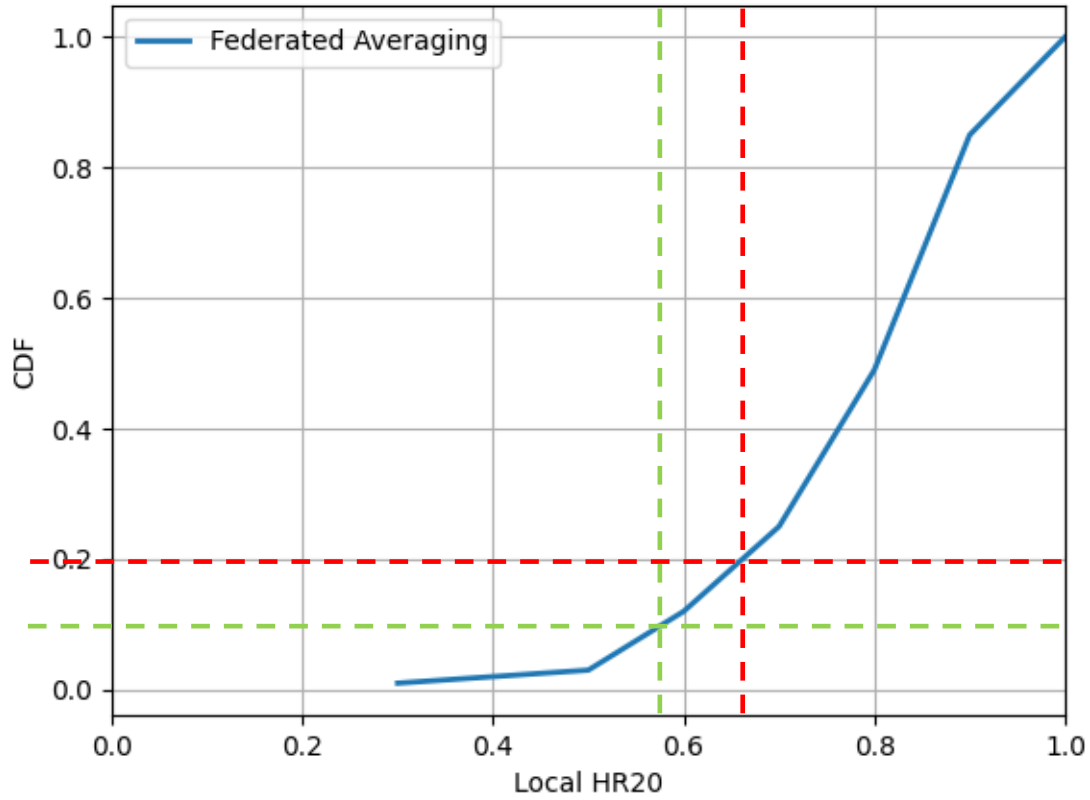


Gossip Learning

- Properties
  - Removes the trust assumption on a central entity
  - Removes the central point of failure
  - Scales better with the increasing number of clients
- But
  - Model convergence (network connectivity, dynamics, device heterogeneïty)?
  - Privacy (attack surface increased or reduced)?
  - Resilience to malicious clients? Selfish clients?

Personalisation Challenge

- MovieLens dataset: 1000 users
- Model: GMF
- Metric
  - HitRatio20 computed at each node
- Average HitRatio20: 80%
- But: clear head and tail users can be distinguished



Decentralized RecSys: focus on personalisation

- How to improve users' local satisfaction?
- Two protocols:
  - Peer sampling
    - Personalized peer-sampling service
  - Model aggregation
    - Performance-based aggregation function

Performance-based Aggregation

- Use a local validation set to evaluate the received models compared to the local model
- Aggregate the local and received model weighted wrt their performance

M3 > M2 > M > M1

M : Local model

Performance-based Aggregation

Algorithm 2: Performance-based Aggregation Function

Data: Local Dataset D_i, currentModel M_i, $WeightingSize$, $TestSize$

Procedure INIT:

- $D_i^{Weighting} = RandomlySamplesSet(D_i, WeightingSize)$ ▷ Randomly samples the weighting set
- $D_i^{test} = RandomlySamplesSet(D_i \setminus D_i^{Weighting}, TestSize)$ ▷ Randomly samples the test set
- $D_i^{train} = D_i \setminus (D_i^{Weighting} \bigcup D_i^{test})$

Procedure PERFORMANCEBASEDAGGREGATIONFUNCTION(M_x):

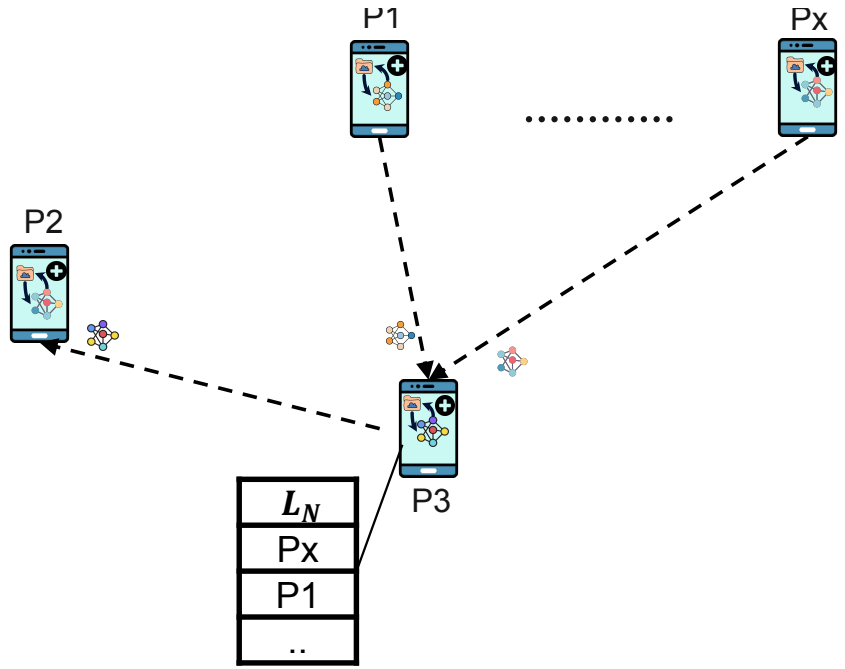
- $P_x = PredictAndEvaluate(M_x, D_i^{weighting})$ ▷ Make prediction with M_x and evaluate its performance
- $P_i = PredictAndEvaluate(M_i, D_i^{weighting})$ ▷ Make prediction with M_i and evaluate its performance
- $M_i = \frac{1}{P_i + P_x}(P_i \times M_i + P_x \times M_x)$ ▷ Aggregate M_i and M_x wrt their performance
- $Update(M_i, D_i^{train})$ ▷ Train the new M_i on local data

Random Peer Sampling in a Nutshell

- Each node has a set of randomly selected neighbors (a view)
- Periodically, each node selects a node in its view and shuffles part of its view with the view of the selected node
- Multiple flavors of Random Peer Sampling protocols exist
  - Dissemination properties (according to the size of the view, shuffling protocol, etc)
  - Resilience properties (to churn, to Byzantine nodes, etc)

Personalized Peer Sampling

- Keeps track of the best received models.
- Considers their owners when sampling peers.
- Also considers random neighbors
  - Exploration/exploitation ratio α.
  - Random Peer Sampling: Exploration dominant strategy.



Evaluation Setup

- Use cases
  - Movie recommendation
  - Point-of-interest recommendation
- Omnet++ simulations, 1000 users
- Models
  - Generalized Matrix Factorization
  - PRME-G
- Competitors
  - Federated (FedAvg, FedFast[SIGKDD'20], Reptile[VLDB'21])
  - Decentralized (Model-Age-Based[JPDC'21], Decentralized FedAvg, Decentralized Reptile)
- Datasets

Dataset	Type	Users	Locations/Movies	Records
Foursquare-NYC	Points of interest	1083	38333	227,428
Gowalla-NYC	Points of interest	718	32924	185,932
MovieLens-100k	Movies Recommendation	943	1682	100,000

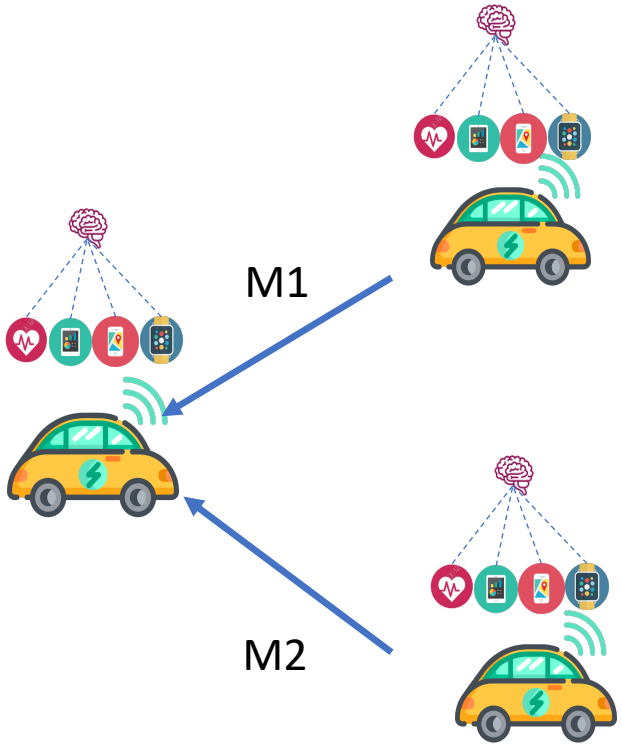
Results

- GMF model
- MovieLens dataset
- Substantial improvement over SOTA solutions (both median and tail)
- More results in the paper

Algorithm\Per-centile HR%	50th	90th	99.9th
Traditionnal ML	40	20	7
FedAvg	33	15	7
Model-Age-Based Method	38	21	0
Decentralized FedAvg	36	21	7
Ours	**46**	**31**	**21**

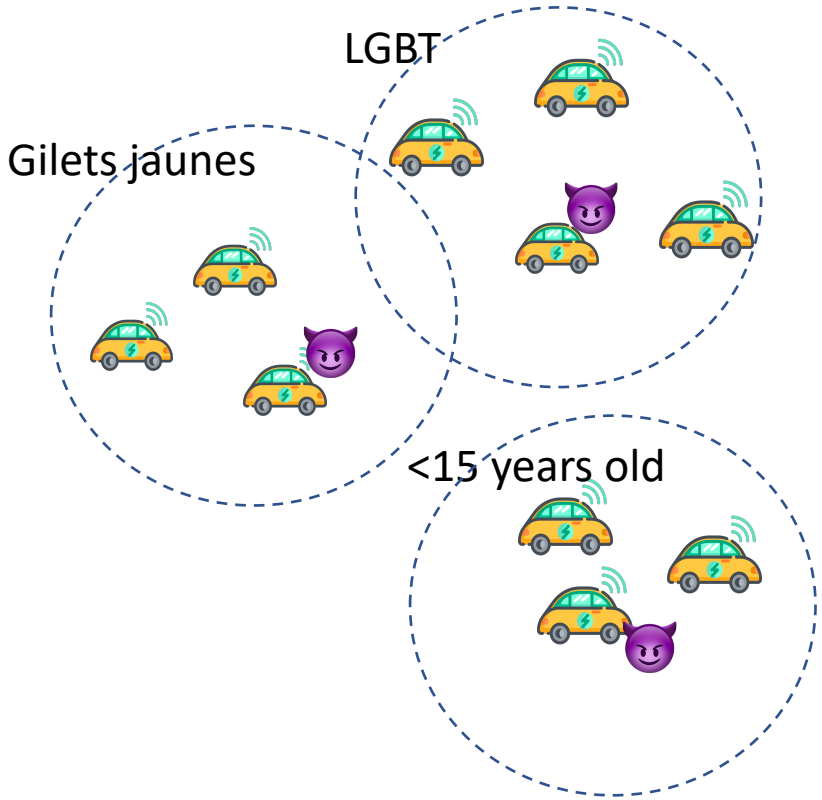
Ongoing/Future Research Directions

- Privacy
  - A node needs to assess the performance of its neighbors' models -> how sensitive?
- Robustness
  - Assess how much performance-based aggregation naturally protects against poisoning attack



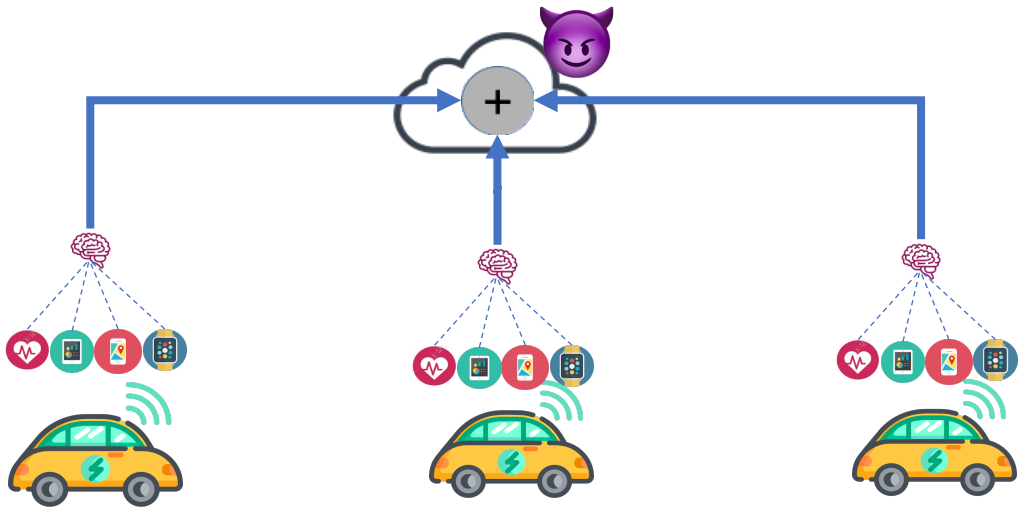
Assessing the sensitivity of model exchanges

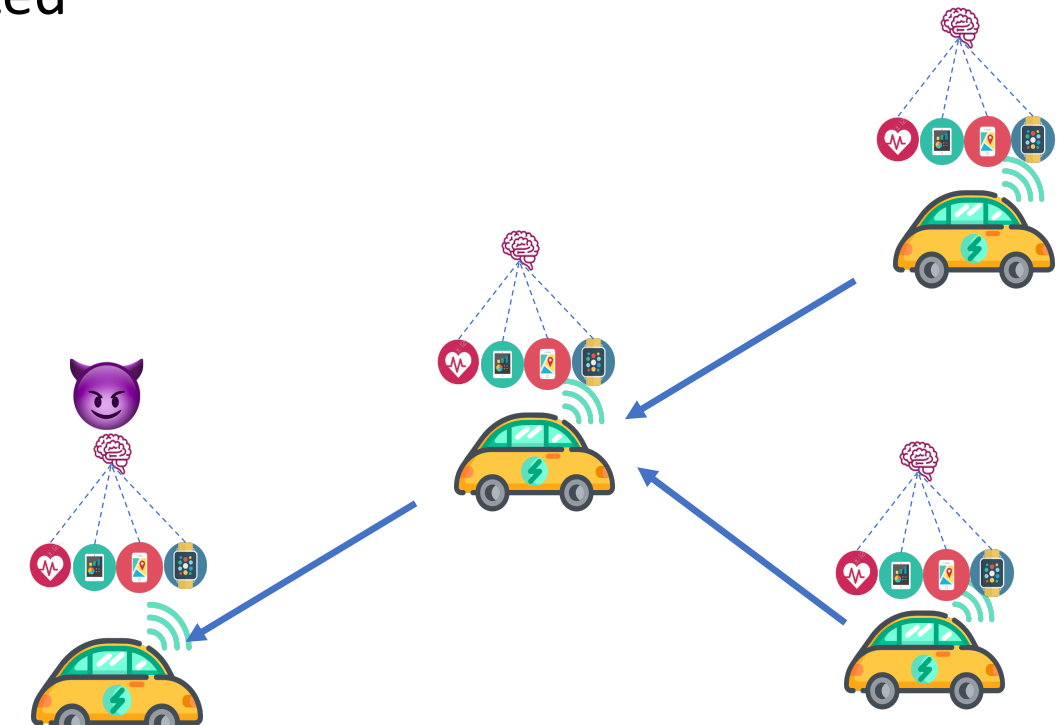
- Can Gossip Learning help an attacker discover communities?
  - All users are honest-but-curious and run the attack
  - Ground truth
    - Off-line computed top-k most similar users for each user
    - Similarity based on rated items
  - Each time a user receives a model, it evaluates the similarity between its locally trained model and the received model and keeps track of its most similar users
  - At the end a comparison is performed between the ground truth and the top-k computed by each user



Impact of Model Dilution on Privacy

- In FL: gradients are “pure”
  - Models are updated with local training only
- In Gossip Learning: gradients are *diluted*
  - A received model might have been aggregated with other users’ models





Conclusion

- Today's online services are too centralized
- A new wave of decentralization is undergoing
- Decentralized ML is needed
- Numerous challenges (ML, optimization, distributed systems/algorithms, security, privacy, networking…)
  - Understand the benefits/limits of decentralization
    - Why did previous decentralization waves fail?
    - Does decentralization increase or reduce the attack surface?
    - Enforcing privacy & resilience to Byzantine nodes: possible?
    - What can we do beyond empirical works?