Cloud Computing

DATA 604

Leanne Wu

lewu@ucalgary.ca
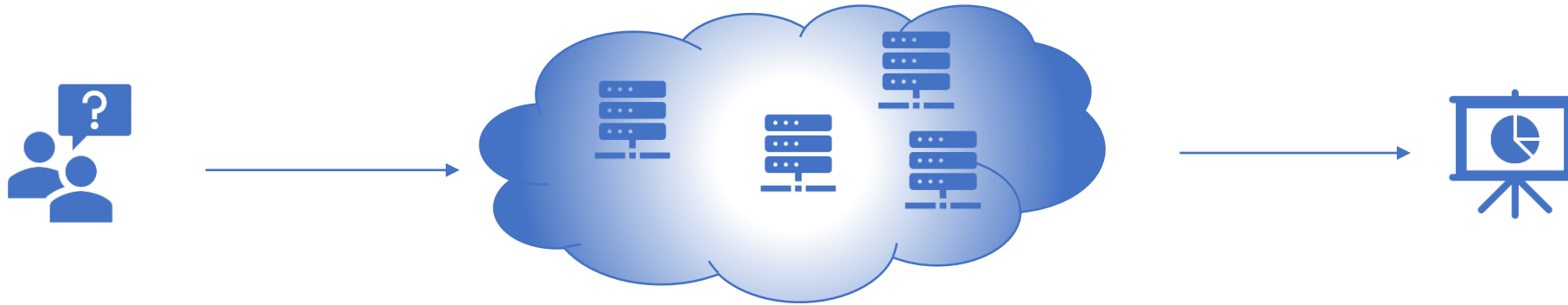
Department of Computer Science

Cloud?

- The idea is relatively simple: provide access to computing located remotely, in such a way that it can be provisioned in a way that is opaque to the user

- Intended to be flexible, lower maintenance costs, share computing resources with other consumers when you don't need them
- Cloud lets consumers scale up computation and storage in ways which previously were not accessible to most organizations

The history of cloud

- Not a new idea
  - previous iterations of the idea included grid computing, utility computing
- Advances in specific technologies paved the way for cloud
  - Virtualization: Logically, users see the same computer, but it can be located anywhere and on any hardware
  - Hadoop/MapReduce: provide a way to automatically coordinate workloads and data across flexible clusters
- Companies which had very large server resources began to look for a way to monetize their spare resources

Who owns the cloud?

- Public: available to any organization who needs to access computing resources (For example: Google Cloud, Amazon Web Services, Azure)
  - Cloud provider will determine how data is managed and provisioned
- Private: owned by an organization who requires flexible computing but does not want to share with another organization
  - Costs of maintenance are high, but integrity of data is easier to maintain
- Hybrid: mix-and-match both approaches to customize appropriate computing solutions

The Internet Of Things

- Cloud computing made it feasible to scale up the use of technologies which collected and required data to function

- IoT infrastructure is generally composed of:

Devices	Controller	Server
Sensors and actuators	Computer	Computer (possibly cluster)
Small but many	Manages devices	Performs analytics
Limited resources	Aggregates data	
Often inaccurate/unavailable		

Internet of Things – illustrated
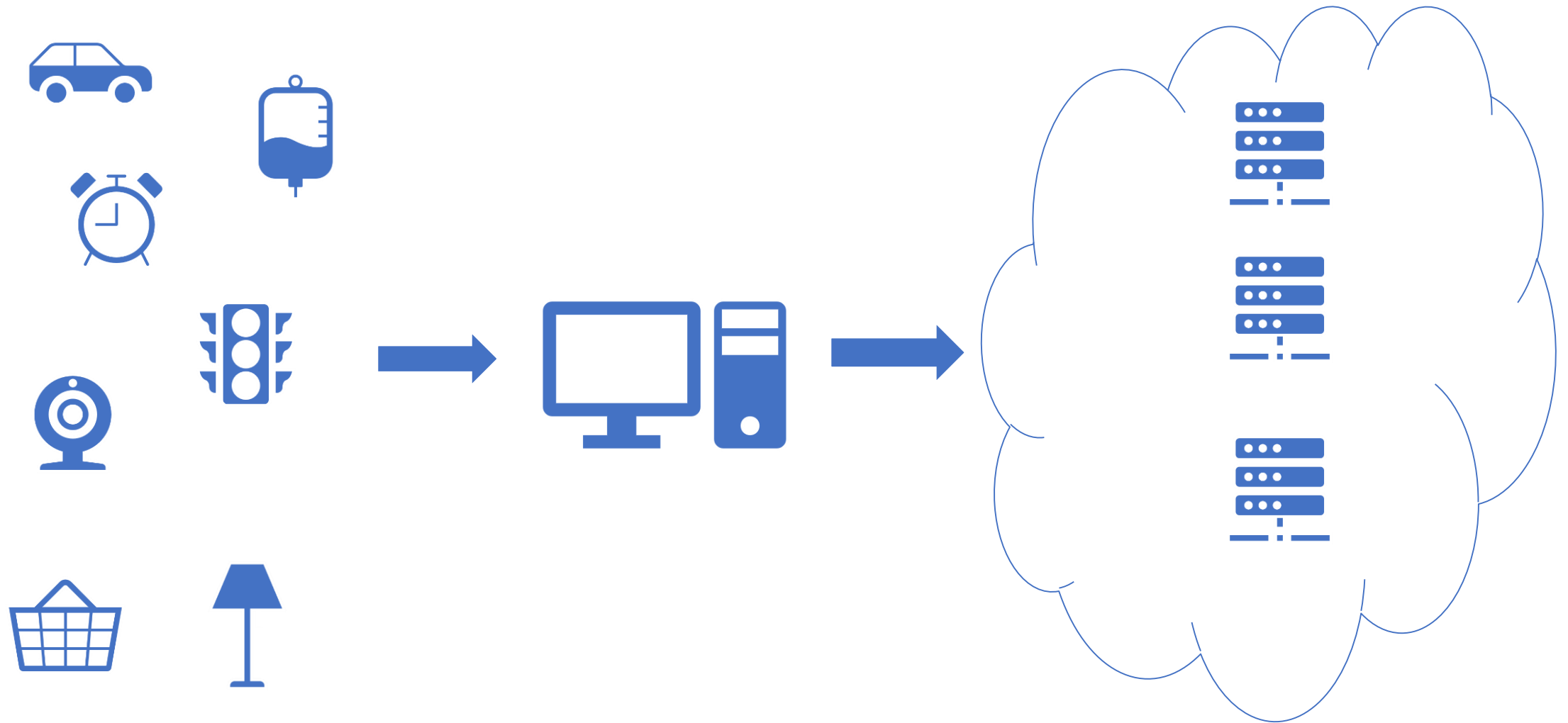
Potential issues

- Reliability
- Security
- Privacy
- Latency
- Long-term support
- Sustainability

Blockchain simulation

- Separate into four groups
- If you receive a quiz with a sticky note on it, you are a **tabulator**

- Put your name on the quiz
- Do the quiz and then trade your quiz with somebody else in your group
- Everybody grade their new quiz. Put your name on the quiz you marked.

This example is adapted from https://blog.codeanalogies.com/2018/04/18/blockchain-explained-by-trying-to-pass-high-school-math-class/

Blockchain simulation (II)

- Hand your quiz to your chosen tabulator, along with one candy. Tabulators will:
  - check your math
  - stack quizzes, sorted by results, so that the quiz with the higher number goes on top
  - look for a token (I will tell you what to look for) in the room
  - show the token to at least two other groups, produce evidence by getting somebody from each group to initial the quiz on top
  - Hand token and quizzes to me

What happened? What does this have to do with blockchains?

- Blockchains are a means for verifying transactions in peer-to-peer networks in a distributed way
  - peers volunteer to validate transactions in return for a small payment offered by whoever needs to validate their work
  - In cases where many peers may validate the same transaction, a tie-breaking mechanism known as a **proof of work** is used
    - usually an unrelated math problem which introduces a delay (typically a problem that is hard to solve but easy to confirm – such as a cryptographic algorithm)
    - as a blockchain grows, the proof of work should grow more difficult
    - once a solution to a proof of work is found, the answer is shared with the rest of the network to verify the solution, until consensus is reached

Is that a blockchain?

- This walks through the process of adding a block to the chain
- The results of the chain are shared amongst nodes, creating a distributed ledger
  - each block has a unique ID (generated by a hash function on the ID of the previous block)
  - thus, the chain cannot change what has already been written

Possible applications

- Blockchains are a potential solutions to store data when
    - The history of an object is important
    - Transactions must be decentralized
    - Transactions occur in an environment without trust
    - Computing resources are abundant