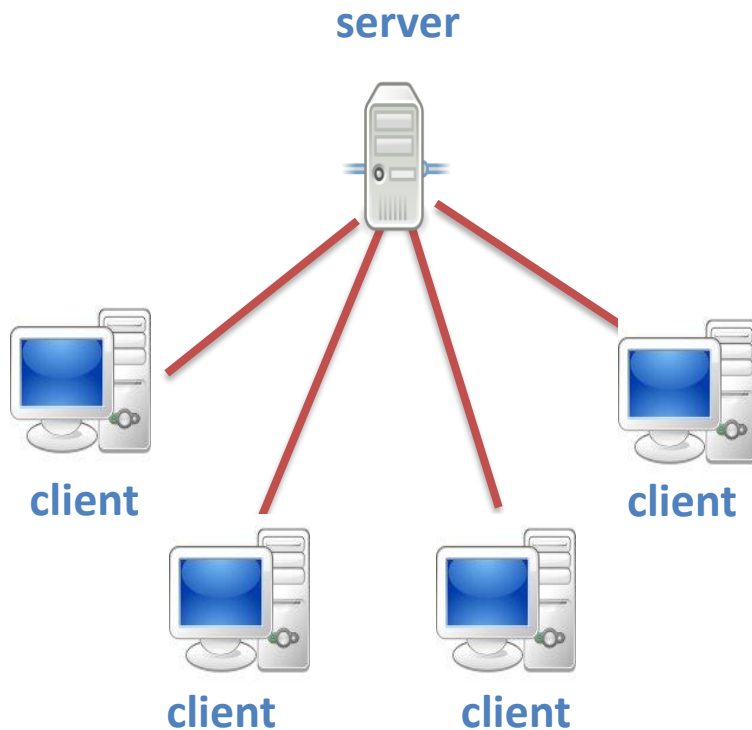


Big Data Analytics

Isara Anantavasilp

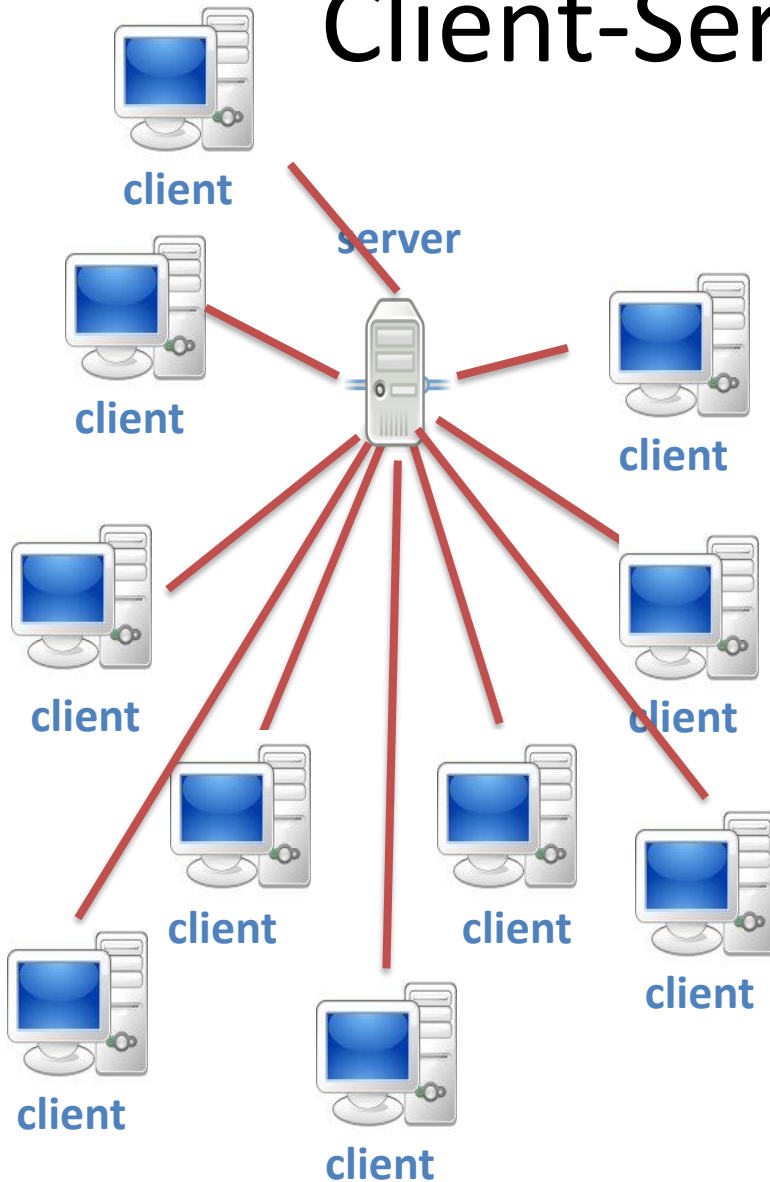
Lecture 4: Hadoop Distributed File System

Client-Server Architecture



- Client-Server is one of the oldest architecture in computer network field
- One powerful machine serves many (weak) clients
- It works!
(WWW, Mail, Games, etc.)

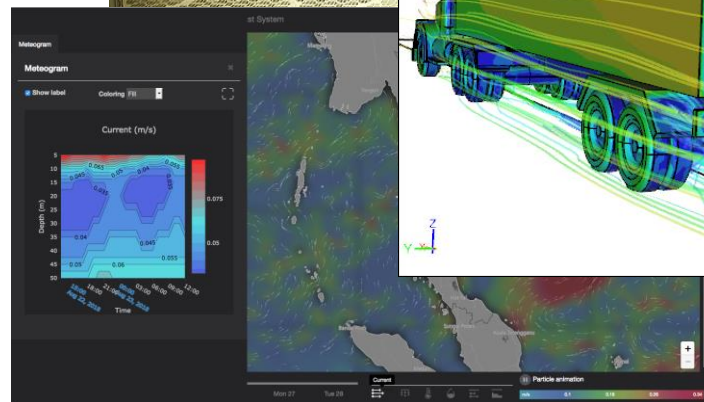
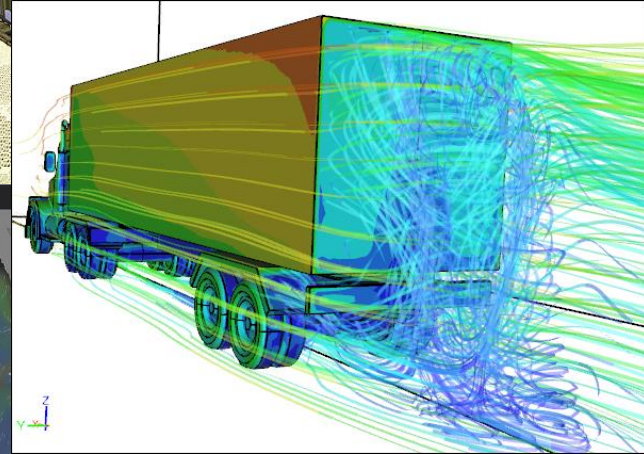
Client-Server Limitations



- This architecture does not scale well
 - Server could be overwhelmed by clients
 - Processing power could be limited
 - Storage could be limited
- Extending processors or expanding drives will not get very far

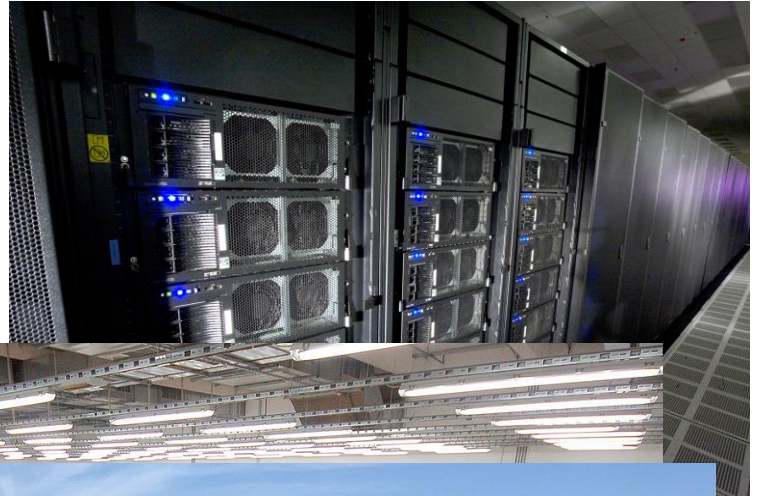
Supersized Server

- Some tasks require **High-Performance Computers (HPC)**
 - Complex task: Weather forecasts, simulations
 - Large data: Search engines, LHC

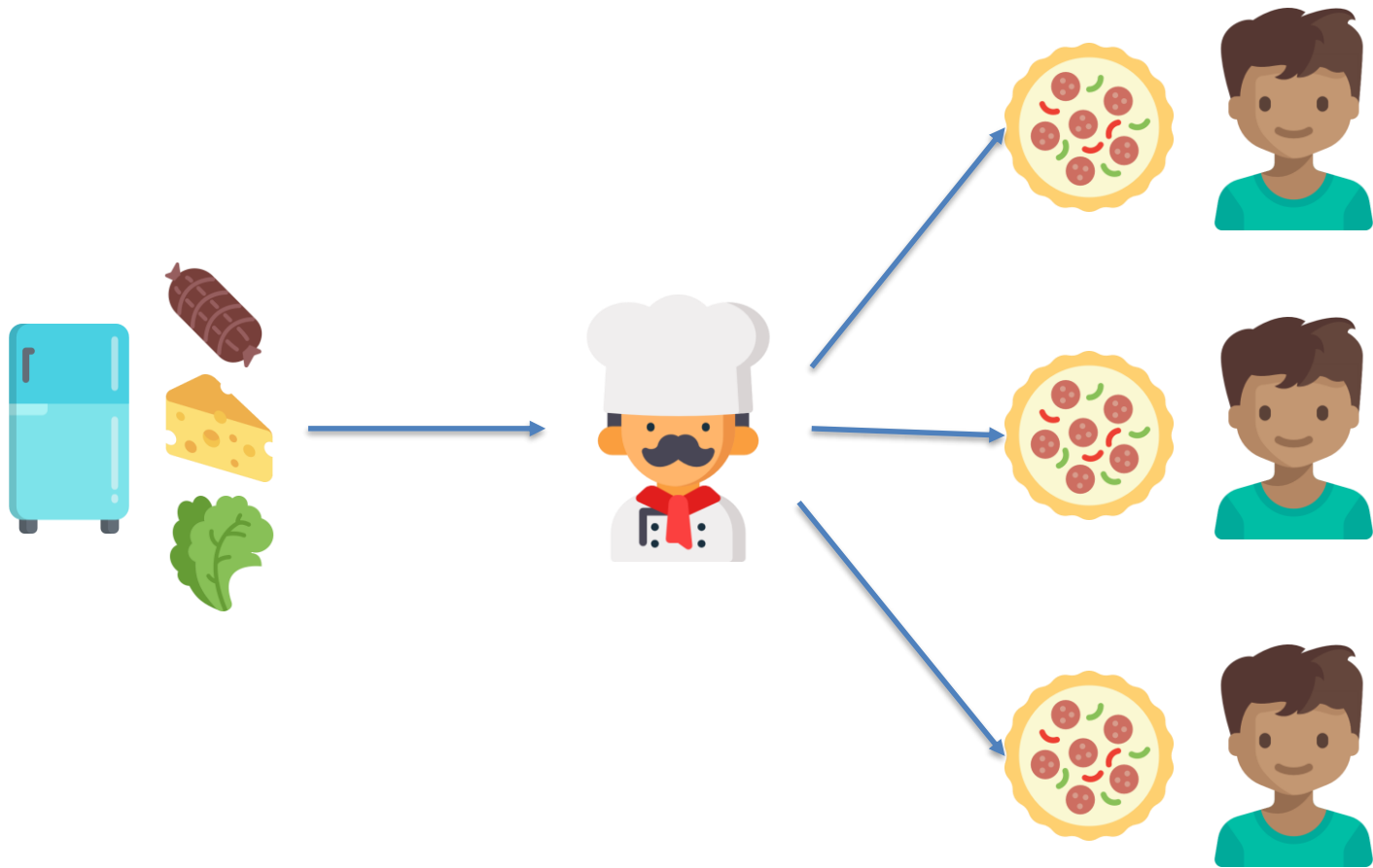


Supersized Server Limitations

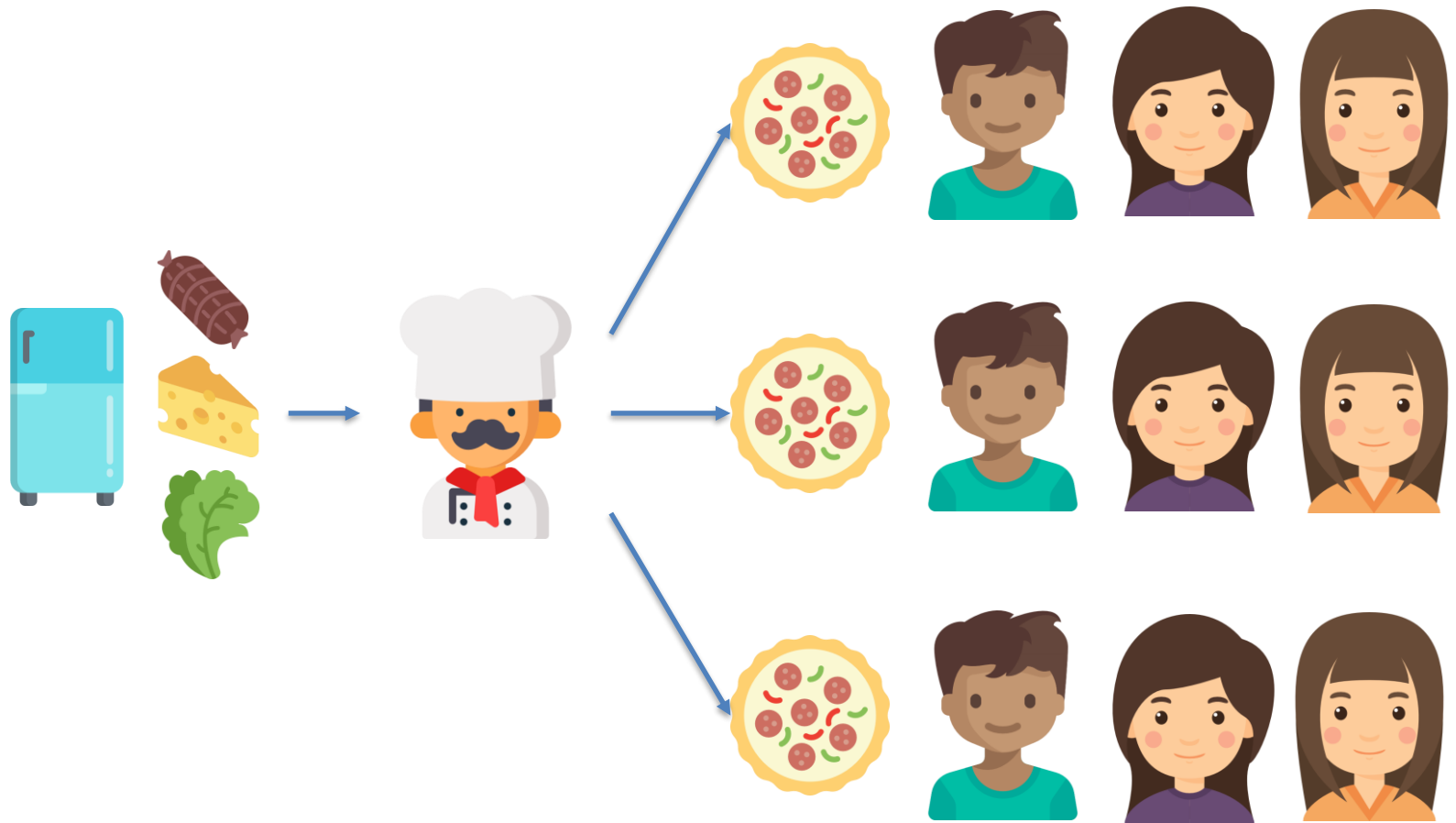
- Expensive
- Specialized hardware
- **EXPENSIVE!**



Rapid Cooking Scenario

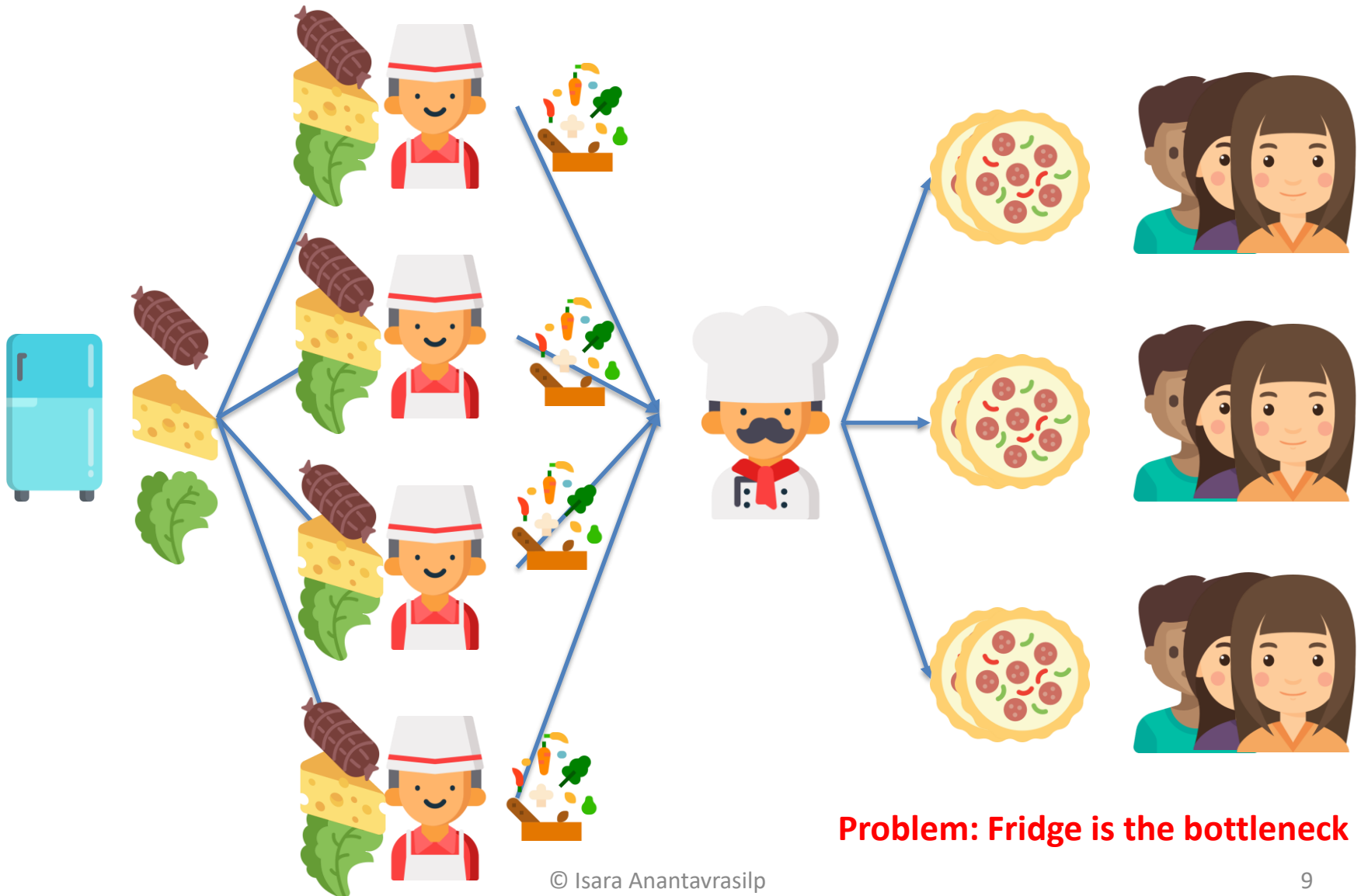


Rapid Cooking Scenario

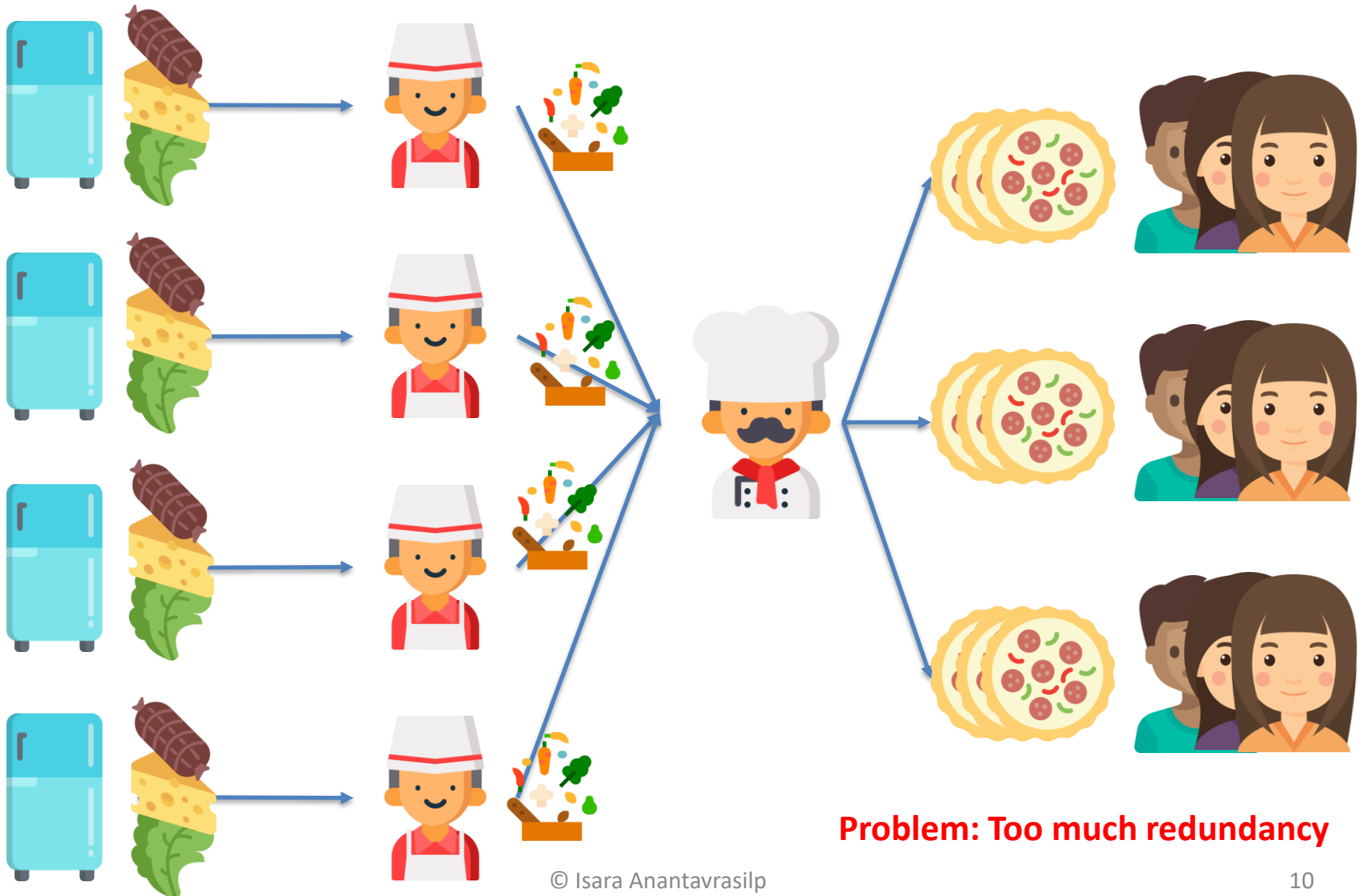


Problem: Does not scale to customers

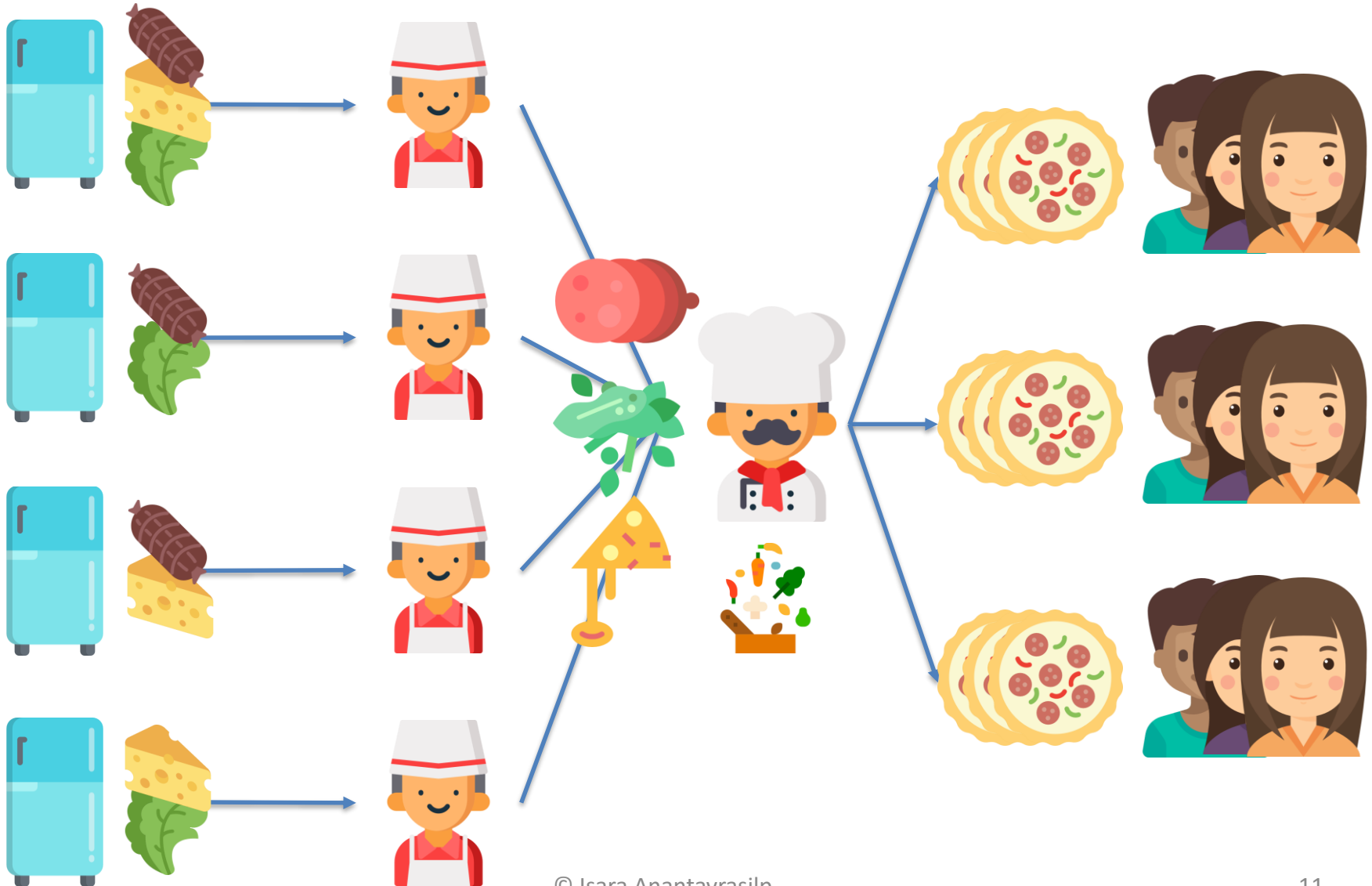
Hiring More Cooks



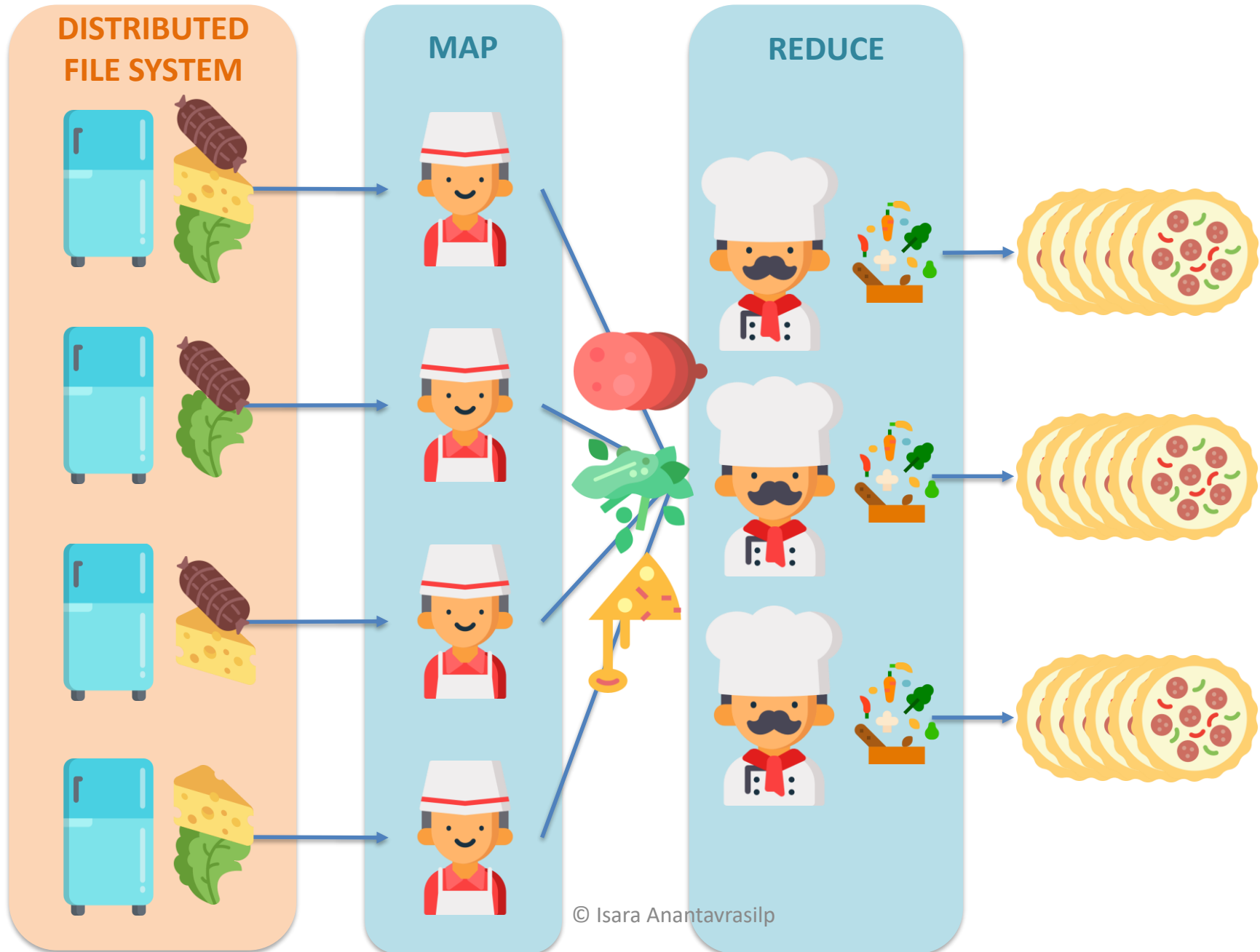
Adding More Fridges



Completely Independent Work



And Scale!





Software library for distributed processing of large data sets using a network of many computers



- Software **library** for distributed processing of large data sets using a network of many computers
 - Composed of many software and components
 - Each component is responsible for different task in data processing and storage
- Designed to store and process large data sets in **parallel** and **distributed** fashion

Hadoop History

- In 2002, Doug Cutting and Mike Cafarella were developing an open source web-crawler called Apache Nutch
- They estimated that to index 1 billion pages, they would need around \$500k for hardware
- However, their architecture would not scale to such volume
- In 2003, Google released a white paper on **Google File System (GFS)**
 - Technology to store large files
 - Distributed file system
- Nutch team implemented their own version called **Nutch Distributed File System (NDFS)**



MapReduce

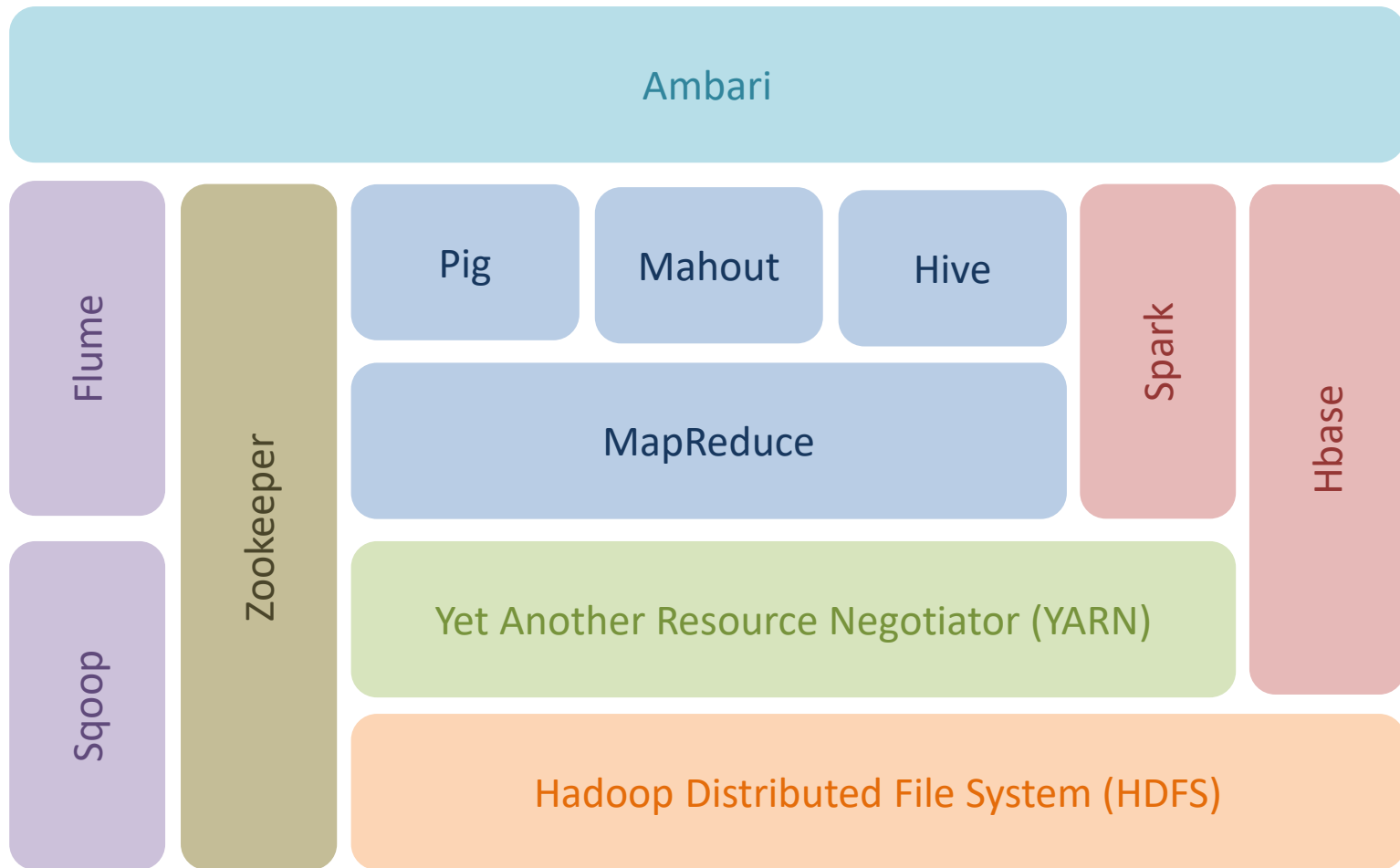


- In 2004 Google introduced **MapReduce** in another paper
- It works closely with GFS
- Nutch team incorporated MapReduce with NDfs
- Finally, they extend the system beyond web crawling and called it **Hadoop**
- In 2008, Hadoop became the fastest system to sort a terabyte of data
- Now, Hadoop has grown very mature

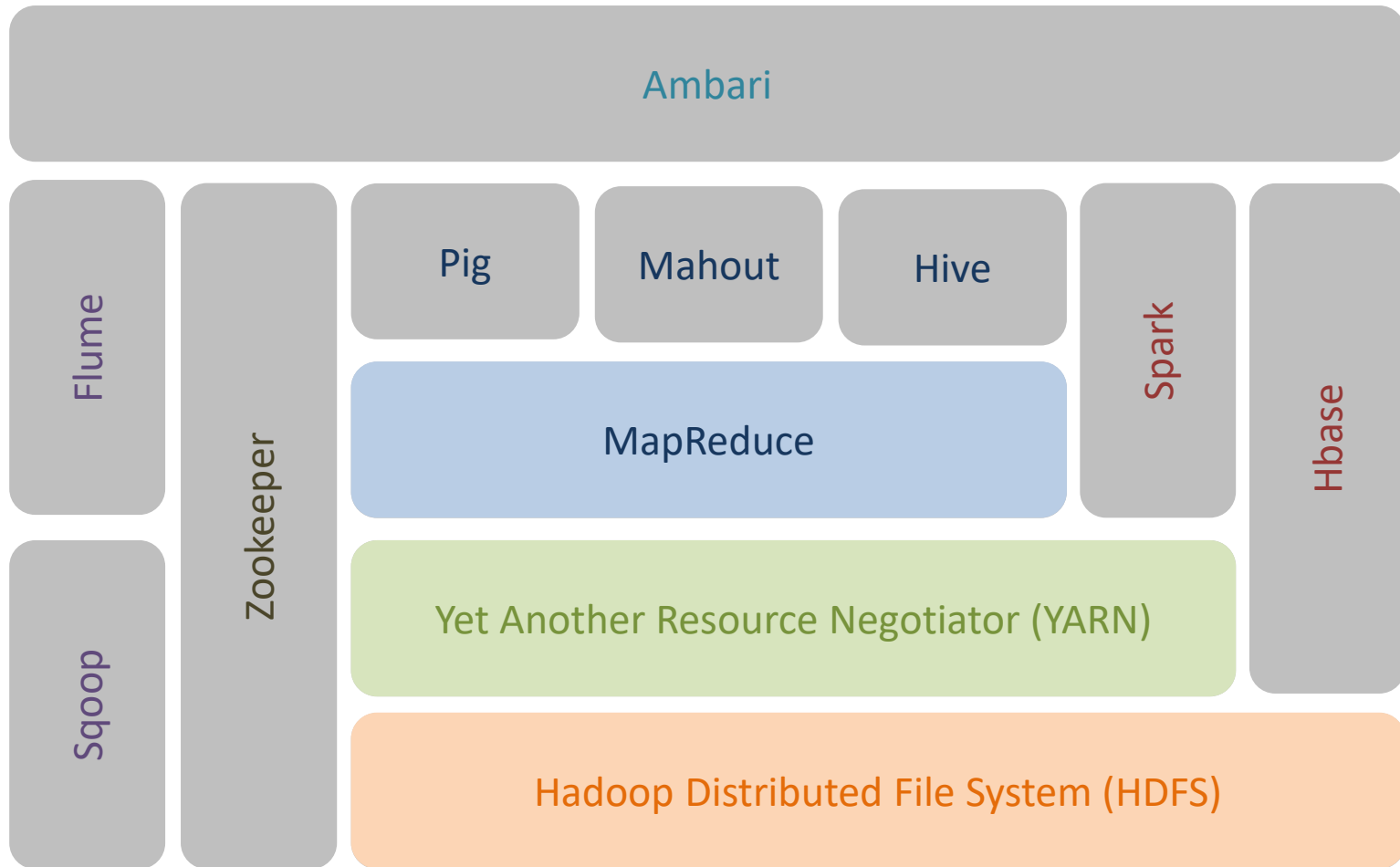
Hadoop Distribution

- **HDFS** takes care of distributed file storage
- **MapReduce** processes the files in distribution fasion
- However there are other operations
 - Query
 - Analysis and processing
 - Resource negotiation
 - Export/import data
- Thus, there are much more components in the **Hadoop Ecosystem**

Hadoop Ecosystem



Hadoop Ecosystem



Hadoop Distribution

cloudera®



MAPR®



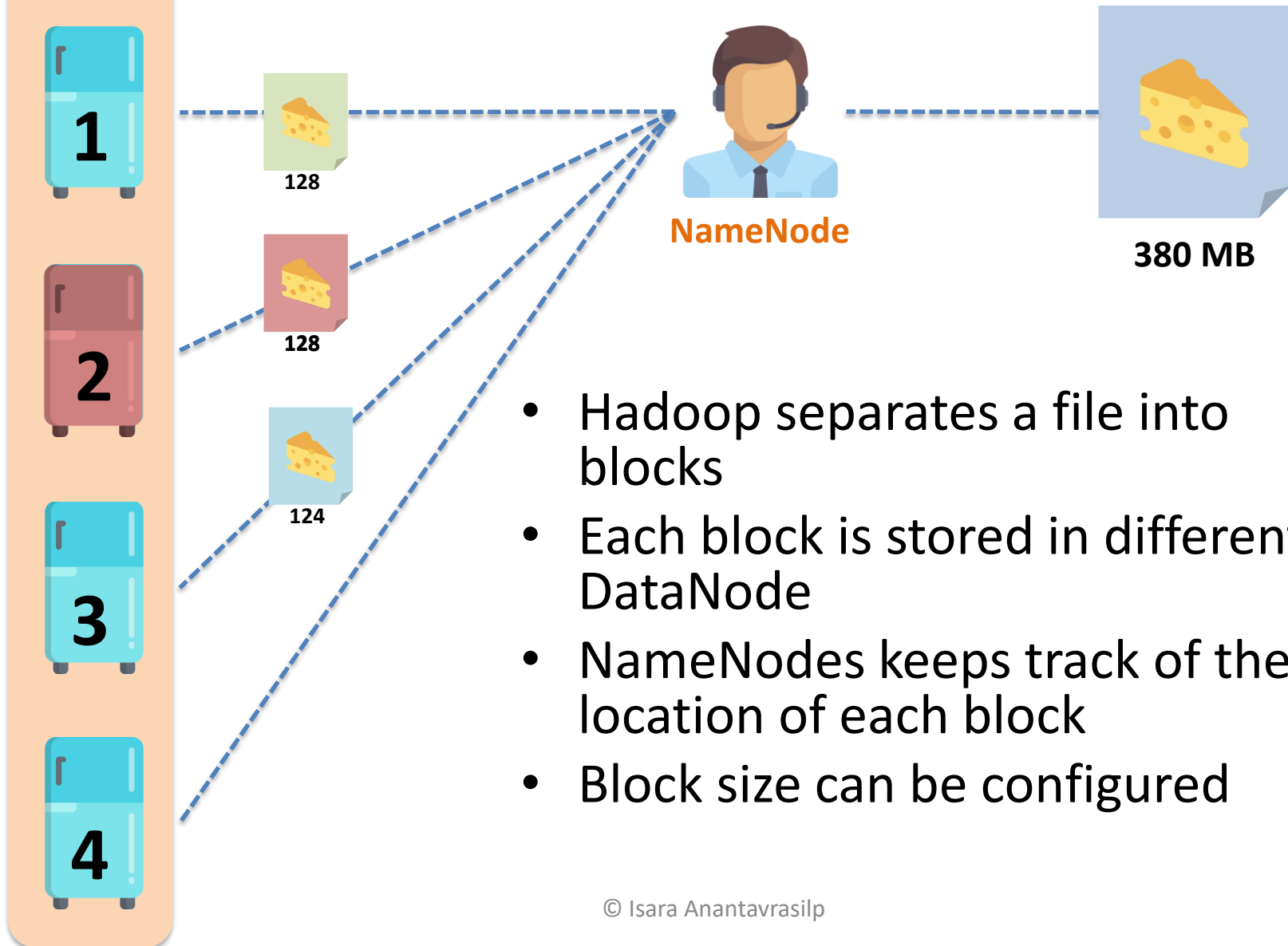
databricks®

Hadoop vs RDBMS

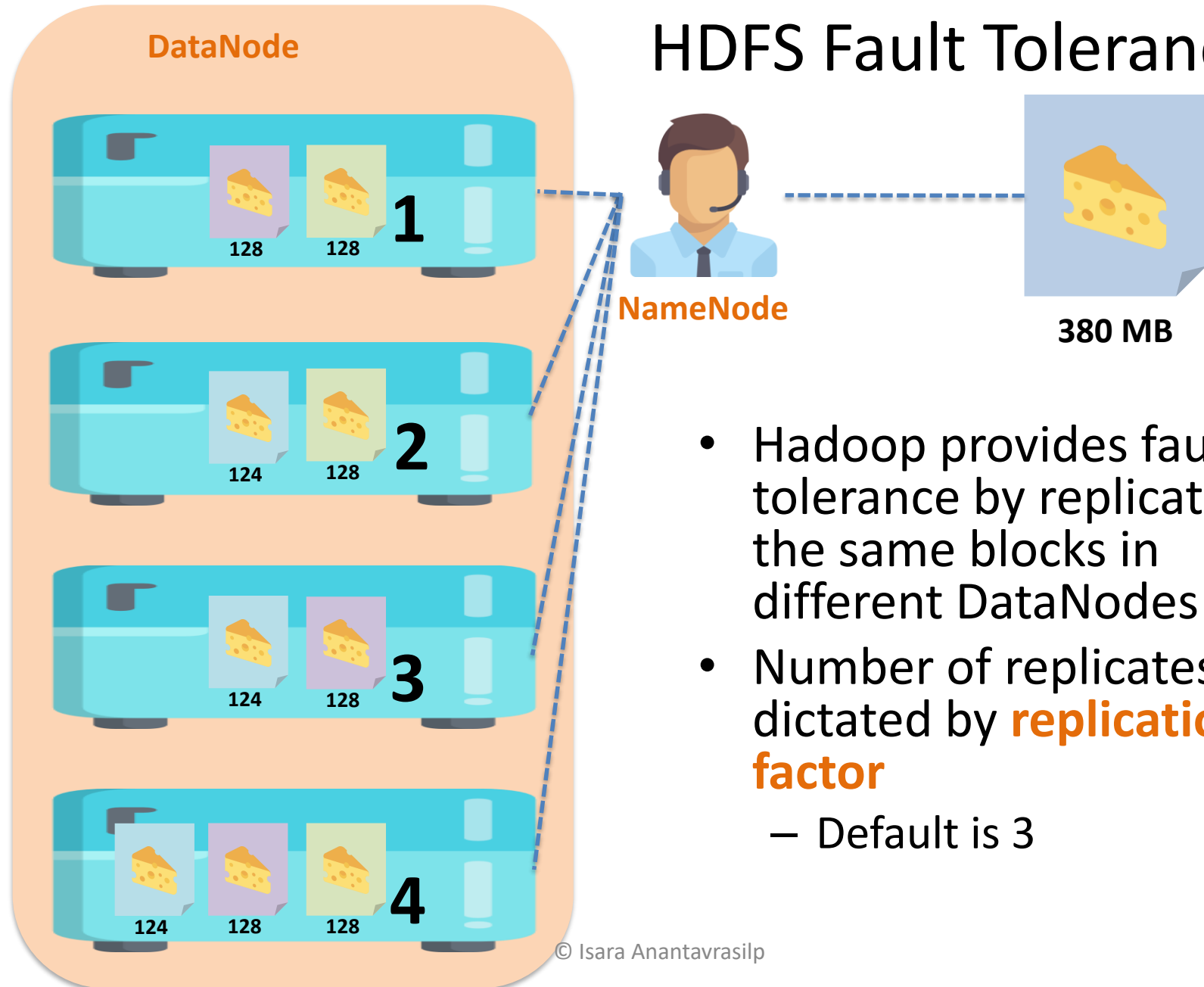
- Hadoop is fast, but it is not for everything
- Hadoop is good for analyzing large files, but not good for small changes
- MapReduce is good for data that are write once read many times.
 - RDBMS is better when the data must be updated often
- MapReduce interpret data while reading (schema on read)
 - RDBMS checks schema at write time

DataNode

HDFS Architecture



HDFS Fault Tolerance



- Hadoop provides fault tolerance by replicated the same blocks in different DataNodes
- Number of replicates is dictated by **replication factor**
 - Default is 3