Decentralized Data Sharing in Web 3.0

Presented by:	Yu Xia	(z5212108)
	Suebtrakul Kongruangkit	(z5176891)
Supervisors:	Helen Paik	
	Xiwei Xu	
Assessor:	Qinghua Lu	

Content Structure

- Overview of the Problem and Project Aim
  - Why Web3.0?
  - Problems of centralized application.
  - Goal of This project.
- Background Knowledge
  - What is SOLID?
  - How blockchain can be useful?
- Proposed Architecture
  - Access control and identity management
  - Architecture for demo application
- Timeline and Future Plan
  - What we've done
  - Thesis B & Thesis C
- Reference

Problem and Project Aim

Different Phrases of Web

- Web 1.0
  - Readable phrase of the WWW
  - Limited interaction between sites and users
- Web 2.0
  - Writable phrase of the WWW
  - Facilitates user interaction
  - Encourages participation, collaboration and information sharing

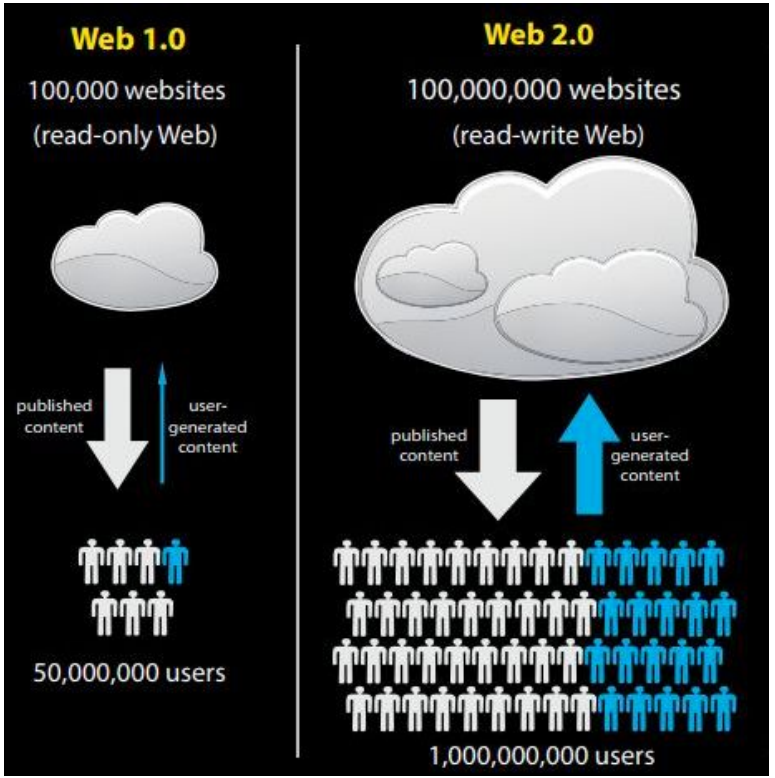
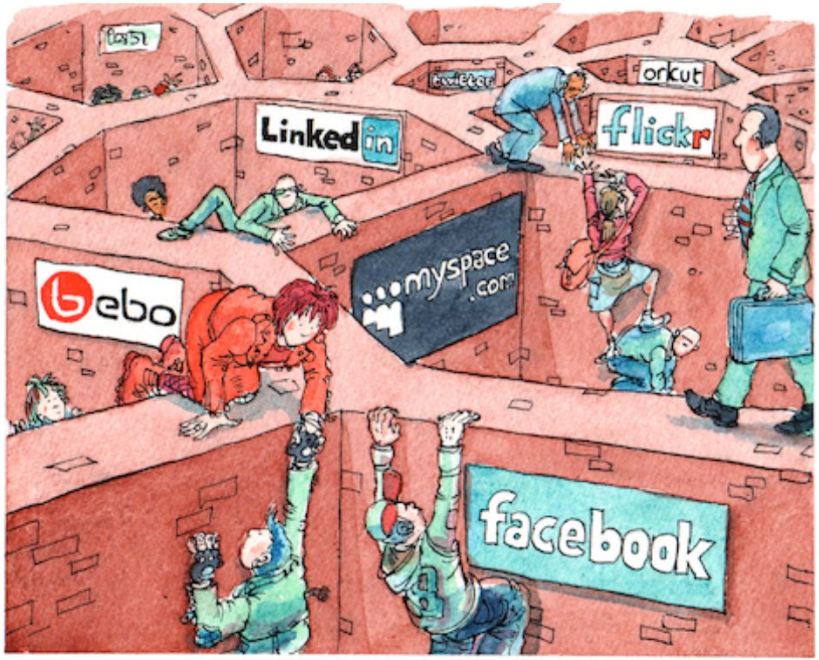


Image Reference:http://researchhubs.com/uploads/web-architecture-4.jpg

Problem of Centralized Website

- Walled Gardens
- Users have no control of their own data
- Lack of data privacy



www.economist.com (illustration by David Simonds)

Web 3.0

- Distributed web
- No longer dependent upon the data center of Web 2.0

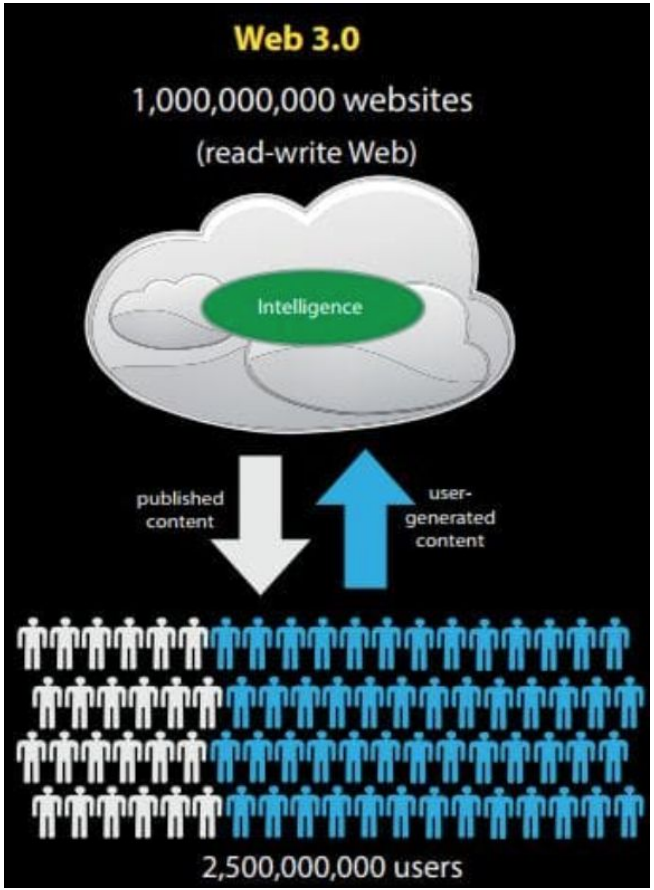


Image Reference:http://researchhubs.com/uploads/web-architecture-4.jpg

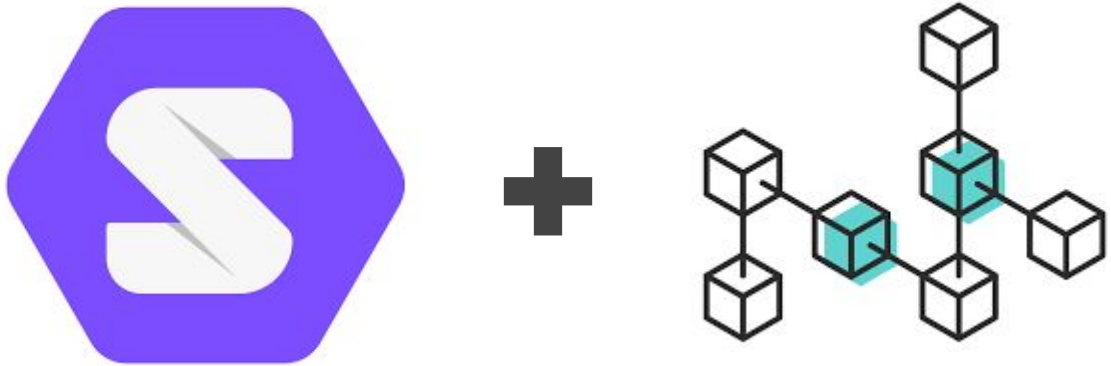
Existing Web 3.0 platform

Polkadot.

elastos

Project Aim

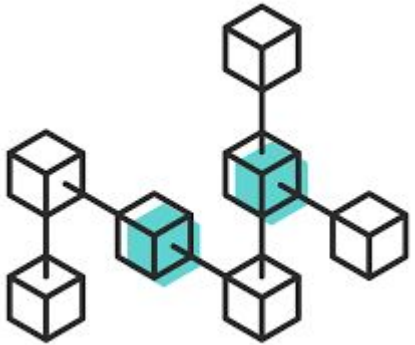
- Investigate, analyze the similarities and differences between 2 existing solutions approaching the idea of decentralized Web 3.0
- Implement the prototype platform that leverages the benefits and idea of 2 communities



Background Knowledge

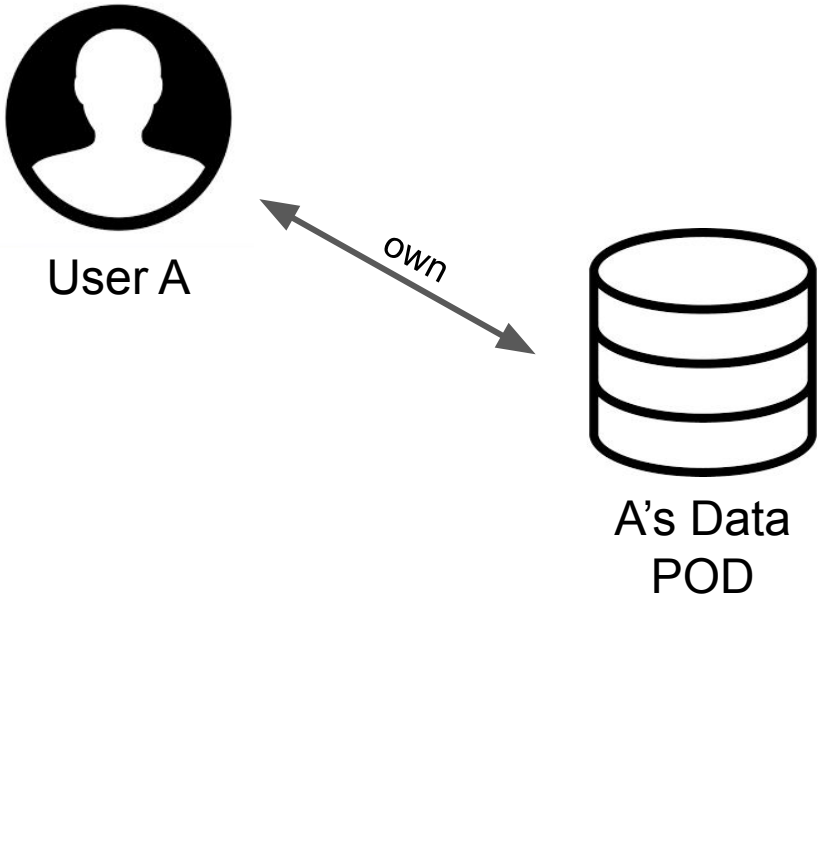
Existing Technologies toward Web 3.0

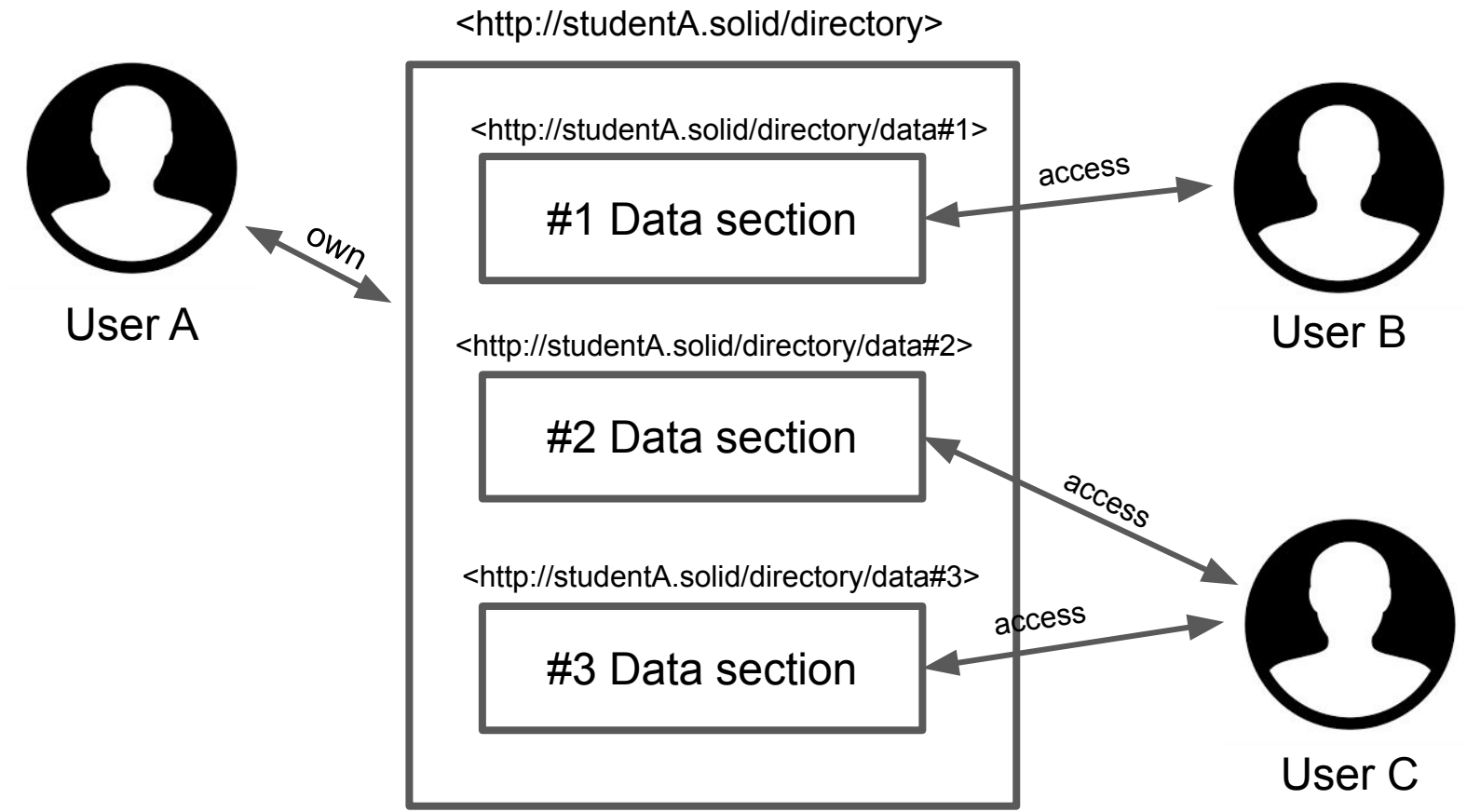
- SOLID
  - Based on existing W3C standards
    - Linked data
    - REST/HTTP/URL based
  - Users have full control of their data
  - Large and active developer community
- Blockchain
  - Completely decentralized network without intermediary
  - Transparent
  - Lack of standardization
  - Smart contract (self-enforceable system)

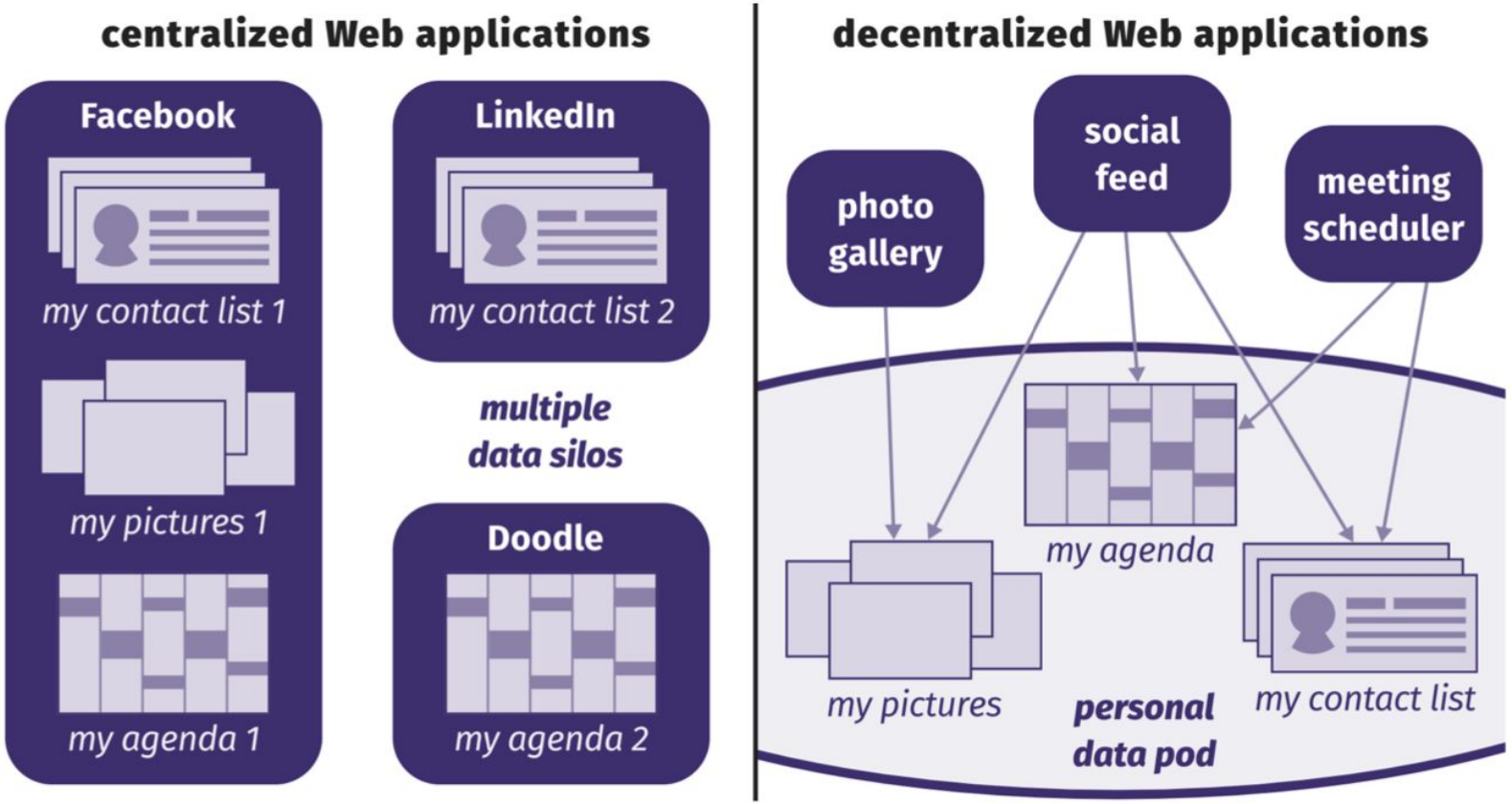


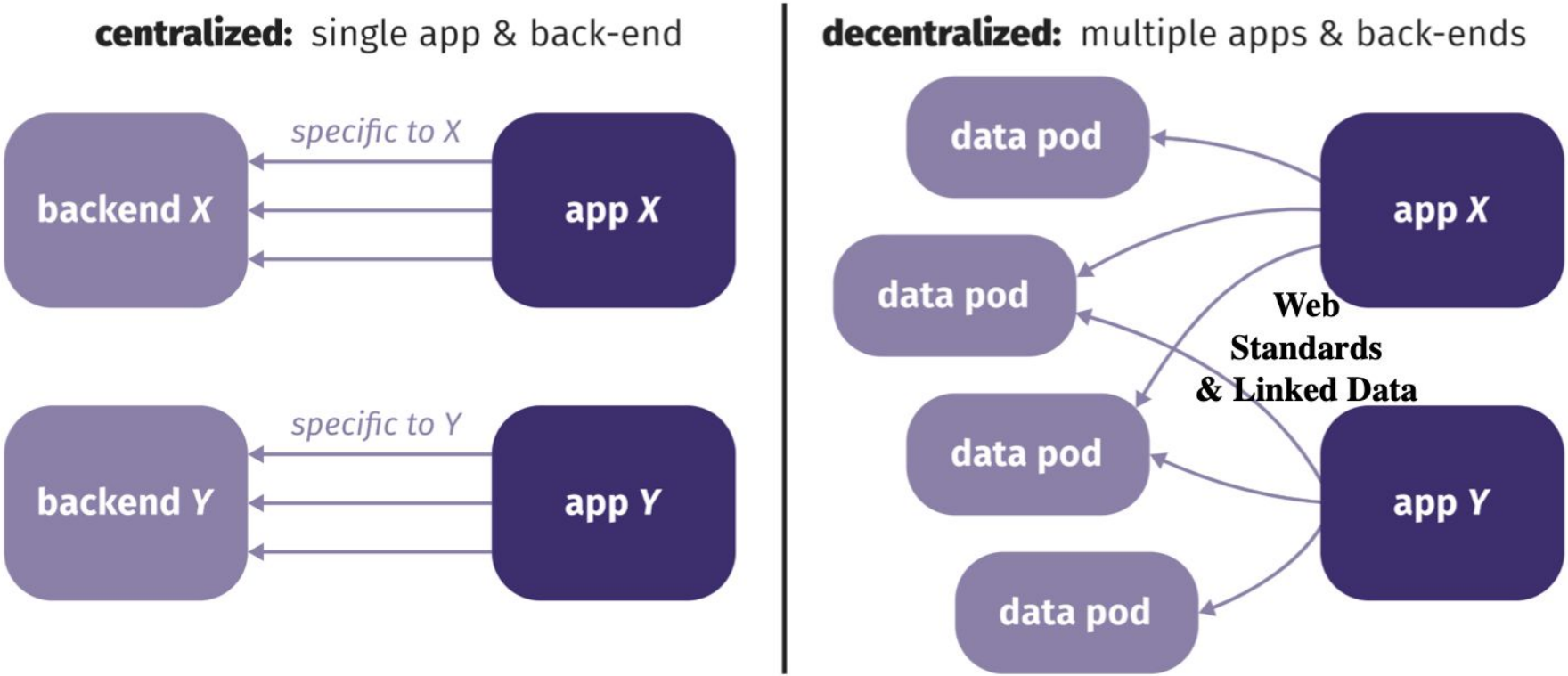
SOLID

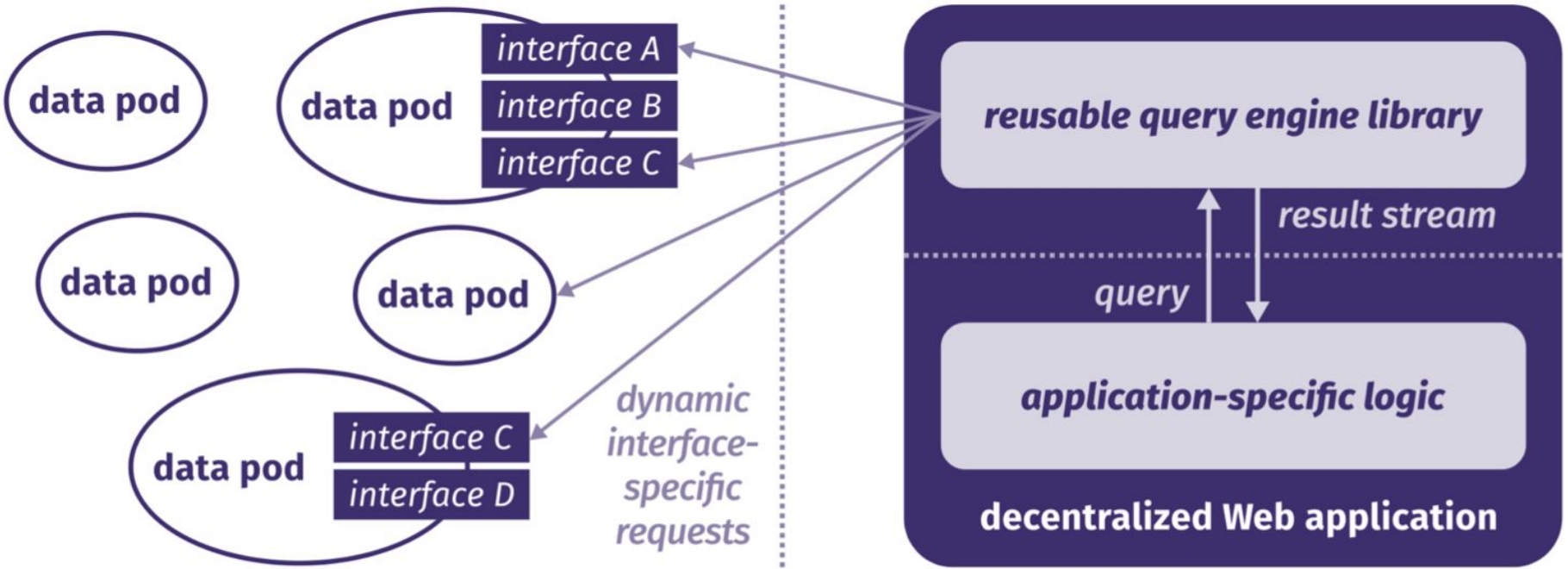
- SOLID: **So**cial **Li**nked **D**ata
- Decentralized Data platform
  - Data are stored inside user personal storage
    - POD: Personal Online Data Store
  - Users have full power to control their data
- Based on existing W3C standard
  - Linked Data
    - Everything is identified by a URL
- HTTP request based access control


User A
own
A's Data
POD
User B
User
C

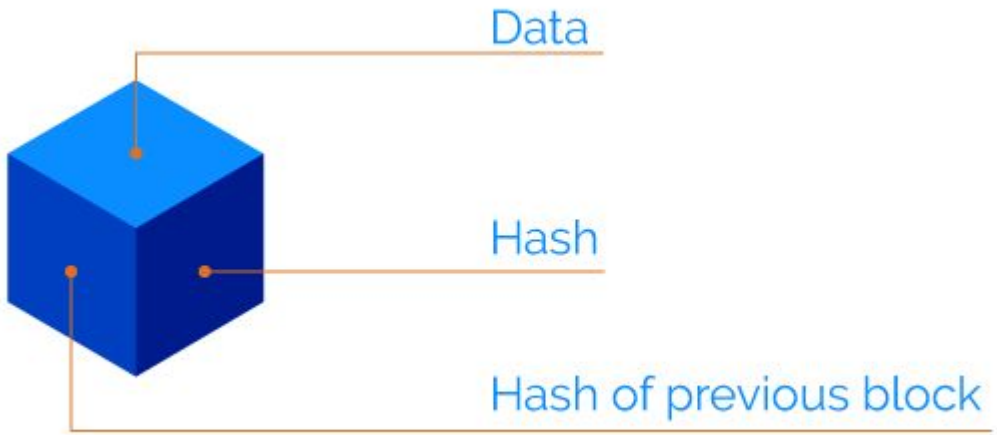

<http://studentA.solid/directory>
<http://studentA.solid/directory/data#1>
#1 Data section
<http://studentA.solid/directory/data#2>
#2 Data section
<http://studentA.solid/directory/data#3>
#3 Data section
own
access
access
access
User A
User B
User C


centralized Web applications
Facebook
my contact list 1
my pictures 1
my agenda 1
LinkedIn
my contact list 2
multiple data silos
Doodle
my agenda 2
decentralized Web applications
photo gallery
social feed
meeting scheduler
my agenda
my pictures
personal data pod
my contact list


centralized: single app & back-end
specific to X
backend X
app X
specific to Y
backend Y
app Y
decentralized: multiple apps & back-ends
data pod
data pod
data pod
data pod
app X
app Y
Web
Standards
& Linked Data


data pod
data pod
interface A
interface B
interface C
data pod
data pod
data pod
interface C
interface D
dynamic interface-specific requests
reusable query engine library
result stream
query
application-specific logic
decentralized Web application

Blockchain



Source: https://rubygarage.org/blog/how-blockchain-works

source: https://rubygarage.org/blog/how-blockchain-works

source: https://rubygarage.org/blog/how-blockchain-works

Smart contract

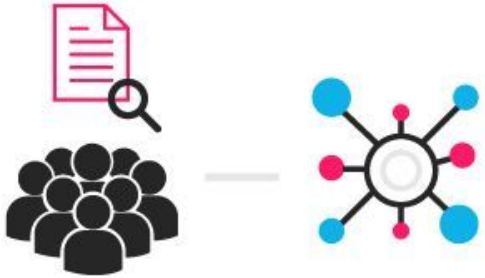
1

An option contact between parties is written as code into the blockchain. The individuals involved are anonymous, but the contact is the public ledger.

2

A triggering event like an expiration date and strike price is hit and the contract executes itself according to the coded terms.

3



Regulators can use the blockchain to understand the activity in the market while maintaining the privacy of individual actors' positions

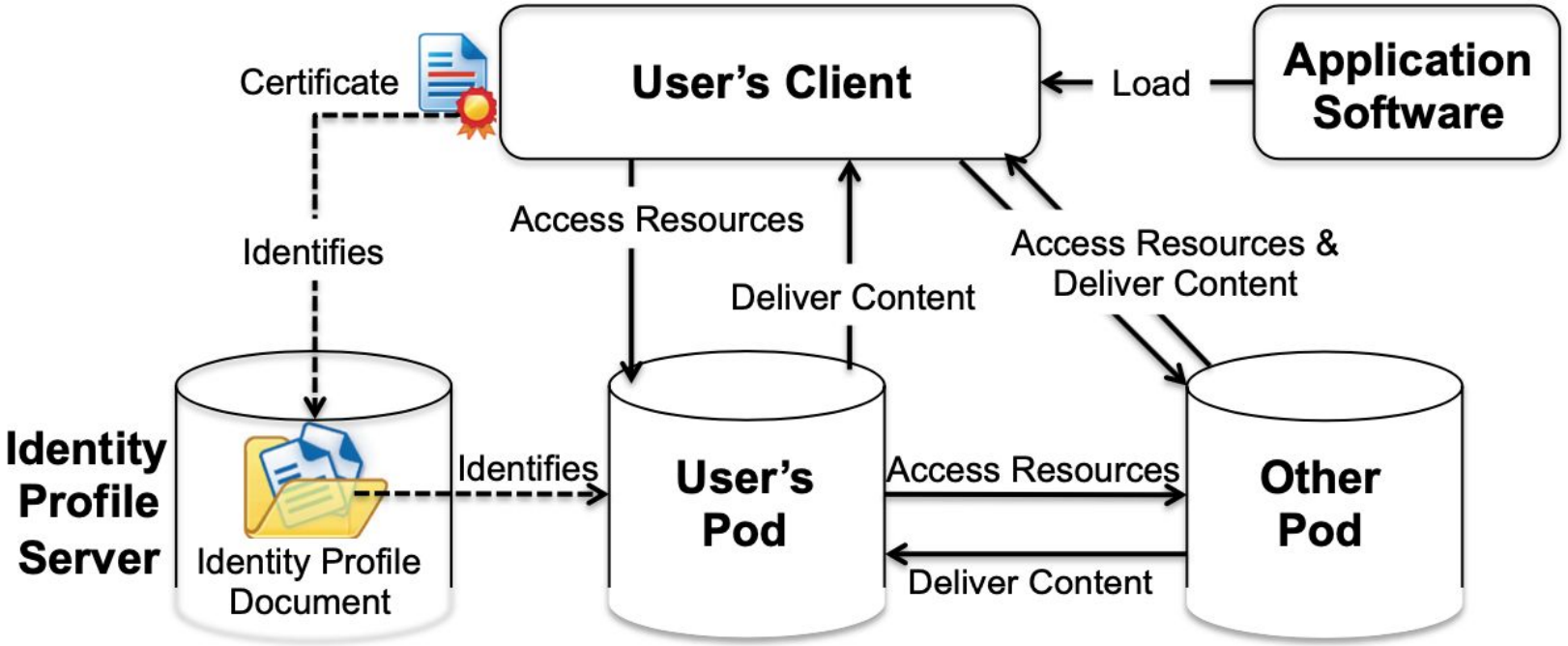
SOLID vs Blockchain

	SOLID	Blockchain

Resource Storage	- Resources are stored inside POD	- Resources are stored on the chain
		  - Smart constract state
		  - Transaction state

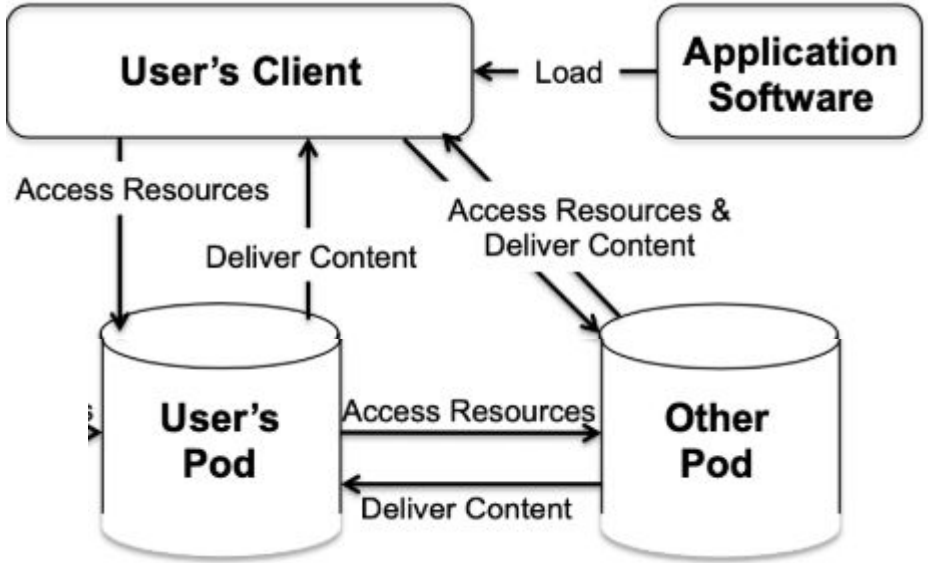
Query	- Using SPARQL and Linked data	- No standardization
Access Control	- HTTP request based for access control	- Can apply smart contract to provide flexibility and automatically enforcing access control policies - Provide data confidentiality and integrity

SOLID Architecture



SOLID Architecture

- User controls identity using RDF profilo document, often stored on a pod server
- User loads a Solid application from application provider
- Application obtains user's pod from the identity provider
- Follow links from the profile to discover data on the user's pod, as well as other pods



Reference:https://pdfs.semanticscholar.org/5ac9/3548fd0628f7ff8ff65b5878d04c79c513c4.pdf?_ga=2.236283139.348460374.1573366957-806205620.1573366957

Proposed Architecture and Application

Proposed Application and Architecture

- We are not aim to build a universal, end-to-end system that somehow magically combines SOLID and blockchain
- We discuss the architecture in the context of applications

Assignment Submission System for Uni Students

- How the pod is managed by individuals?
- How the access control part is used in data sharing?
- What if the university wants to overwrite access control level of students?

Blockchain-based Architecture

- Replacing access control system of the existing one in the architecture

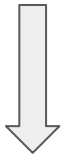
Different Levels of Access Control

- Read, Write, Update, Modify, Append ……

Core Idea

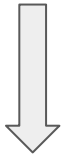
Imagine there exist such an algorithm such that

When A want to share some data with B, C and D



Taking A's key, B's key, C's key, D's key as a source
Encode this data with those four keys

Produce a hash (encoded text)

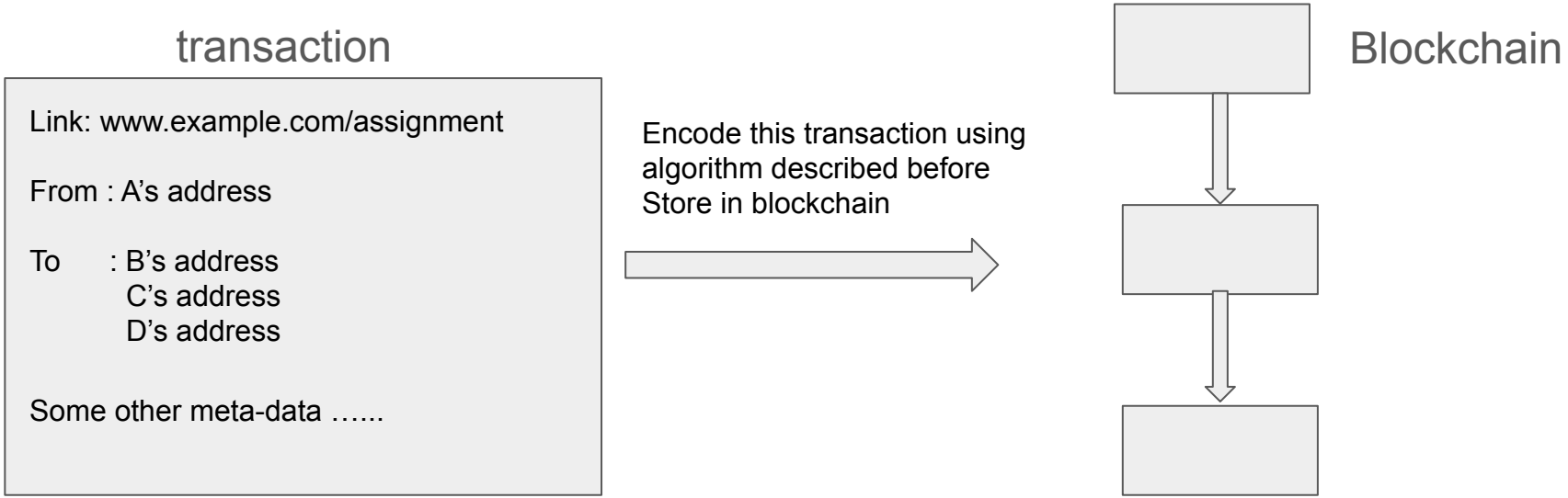


It can only be decoded by one of the keys from the source, e.g, only A or B or or C or D can decode this information. Not anyone else.

Scenario 1

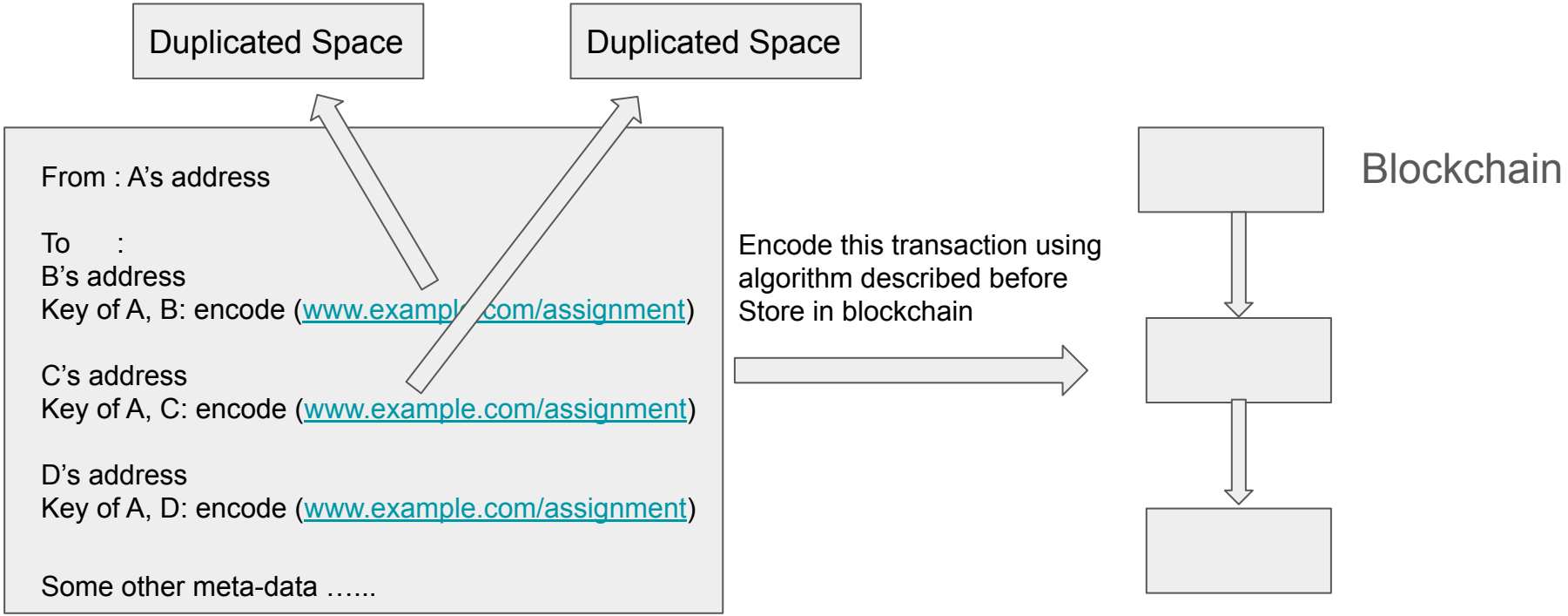
A wants to give “Read” access control of some data to B, C and D.

Originally, the link of that data is private and can only be seen by A.



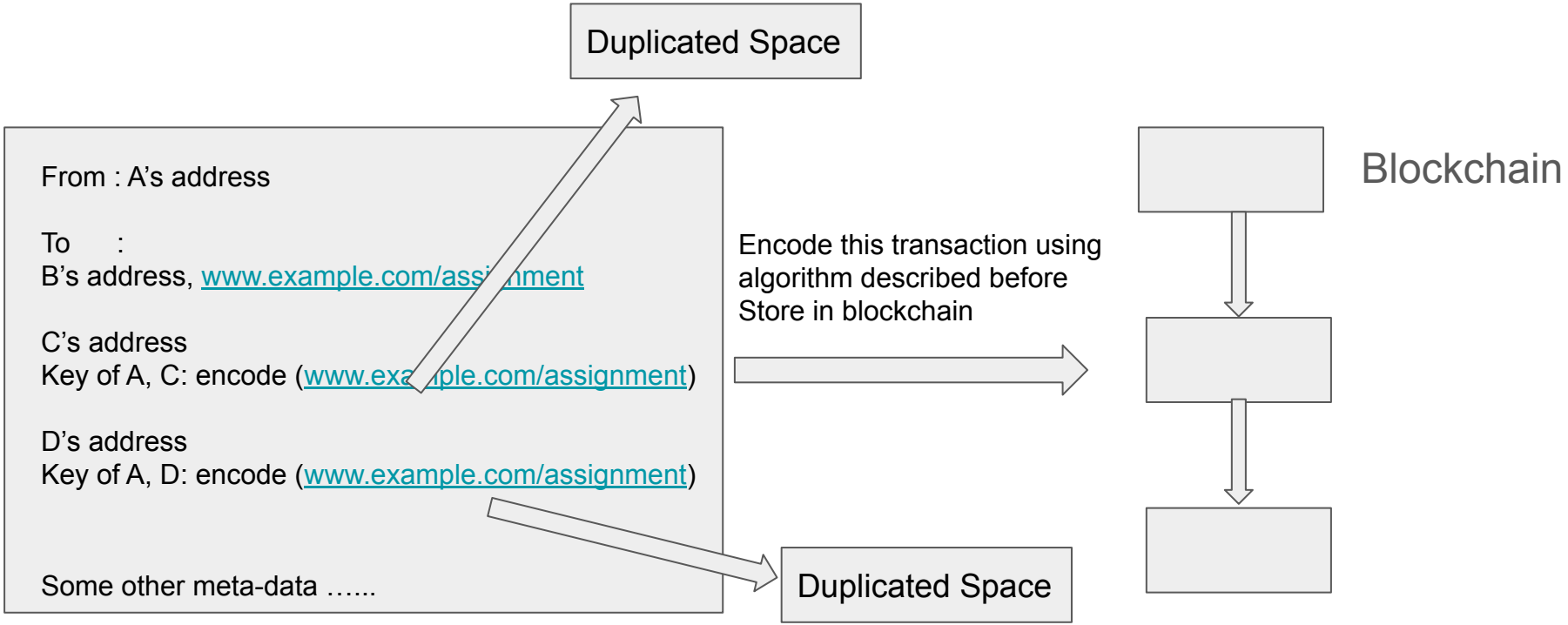
Scenario 2

A wants to give “write access control” of some data to B, C and D



Scenario 3

A wants to give “Read access” to B and “Write access” to C and D



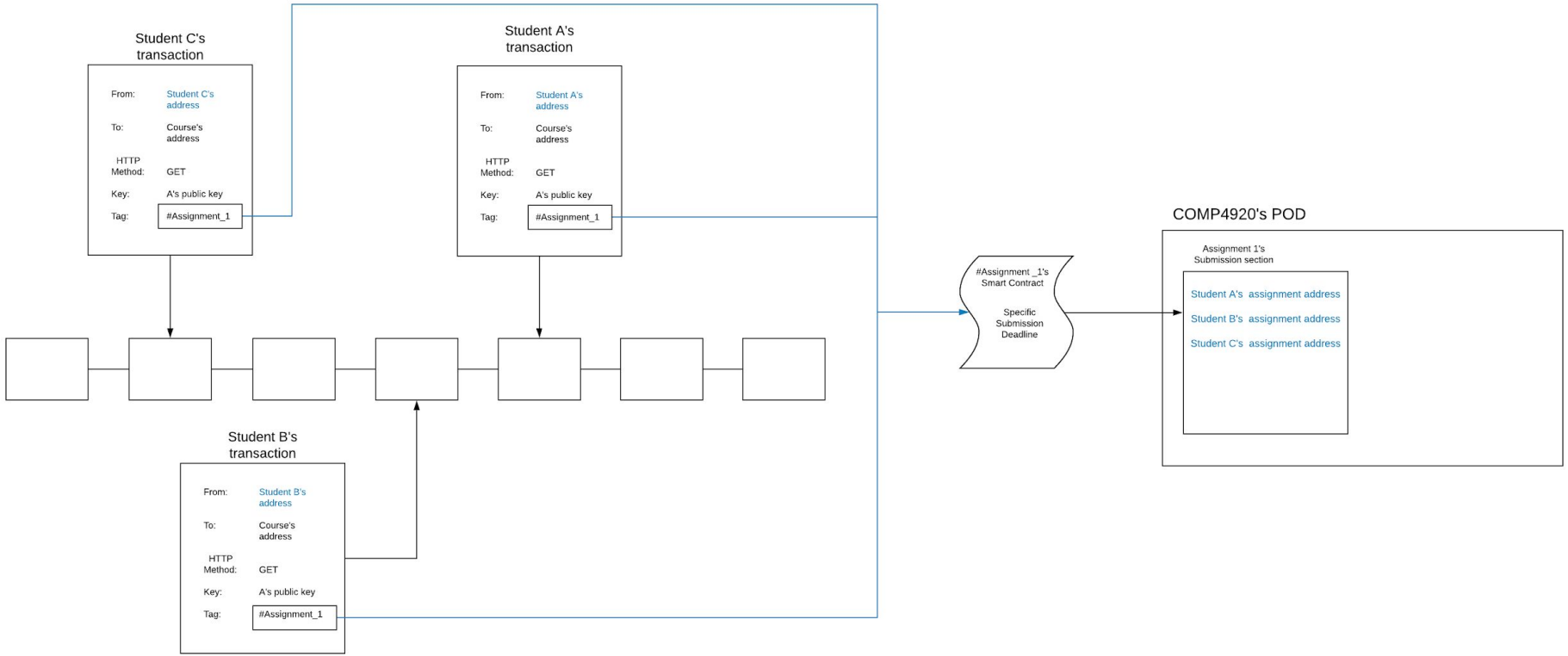
Smart Contract to Enforce Regulations

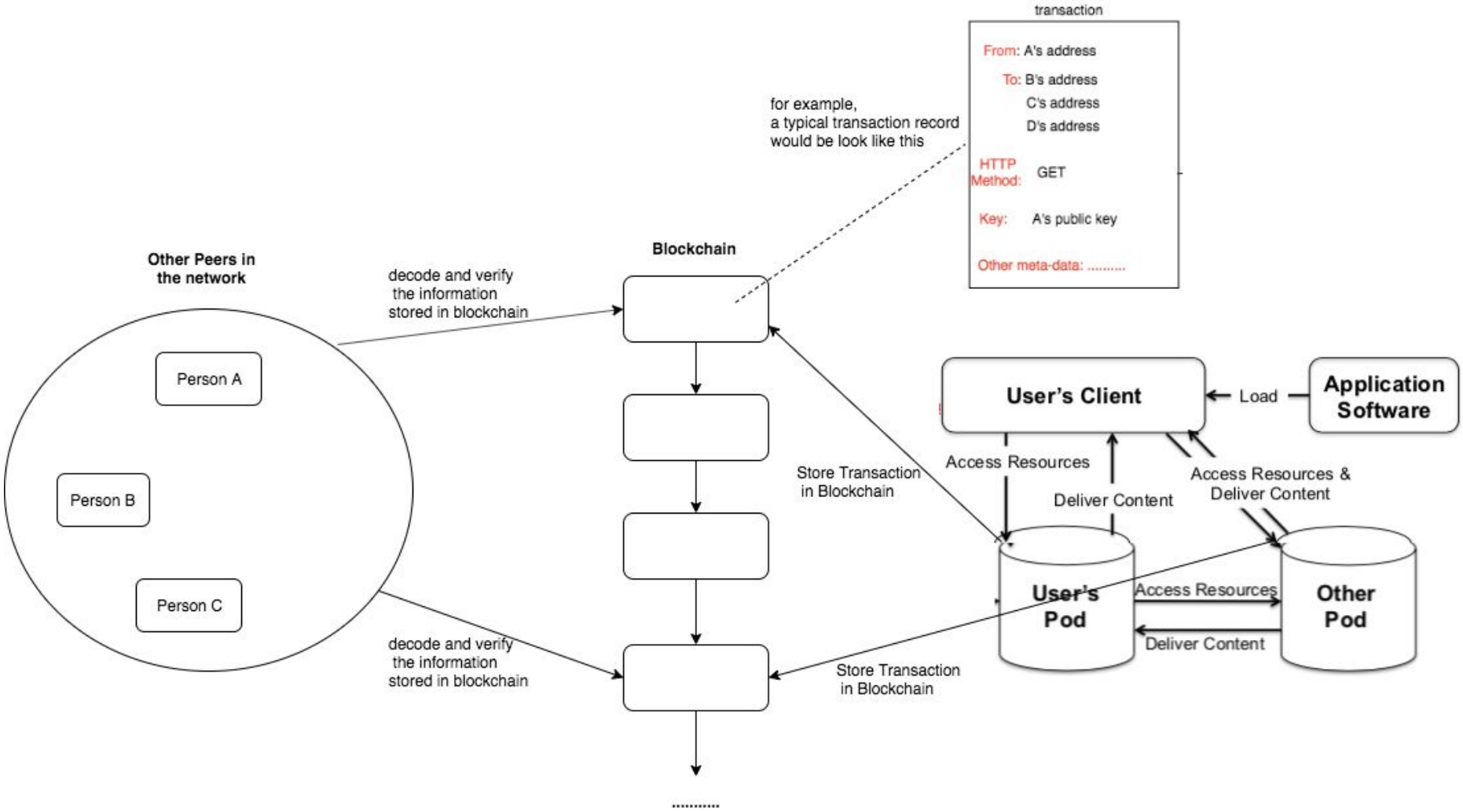
There are some cases that student cannot have full access to their data:

- University force to store students' data for some time
- Cannot submit assignment after 7 or 5 days from deadline
- Cannot share assignment work with other students
- Cannot share solutions from previous year
- ……..

All those can be taken care by smart contract

Resources Submission enhanced by Blockchain




transaction
From: A's address
To: B's address
C's address
D's address
HTTP Method: GET
Key: A's public key
Other meta-data: .....
for example,
a typical transaction record
would be look like this
Other Peers in the network
Person A
Person B
Person C
decode and verify the information stored in blockchain
Blockchain
decode and verify the information stored in blockchain
Store Transaction in Blockchain
Store Transaction in Blockchain
User's Client
Load
Application Software
Access Resources
Deliver Content
Access Resources & Deliver Content
User's Pod
Access Resources
Deliver Content
Other Pod
.....

Timeline and Future plan

Timeline and Future Plan

What we've done so far

- Thesis A
  - Define problems
  - Background research on different approaches
  - Propose rough architecture
  - Presentation and draft literature review (we are at here)

Timeline and Future Plan

Future Plan

- Pre-preparation
  - Read and analysis a few existing SOLID applications' source code
  - Explore APIs that SOLID provides
  - Write user stories for demo application and split tasks
- Thesis B
  - First Half (roughly week 1 — week 6)
    - Build assignment submission application entirely based on SOLID, with existing access control systems
    - Including frontend UI and backend logic
    - Will have a fully functional application on Thesis B presentation
  - Second Half (week 7 — week 10)
    - Refactor code and replace access control and identity management using Blockchain
    - Should be easy to do if we follow good development methodology (MVC)

Timeline and Future Plan

- Thesis C
  - Gain deeper understanding by building the application on SOLID
  - Refine architecture
  - Generalize the access control part to other applications, ont only on assignment submission senario
  - Build more application on this platform if time allows
    - Such as “student resume generating” functionality
    - Got the chance to apply some machine learning and NLP as well, since this is the area we both want to explore

Reference

1. Web 1.0 and Web 2.0 Image: http://researchhubs.com/uploads/web-architecture-4.jpg
2. Problem of Centralized Web Image: www.economist.com
3. Blockchain and smart contract Image: https://rubygarage.org/blog/how-blockchain-works
4. Solid client-side architecture: http://emansour.com/research/meccano/solid_protocols.pdf