

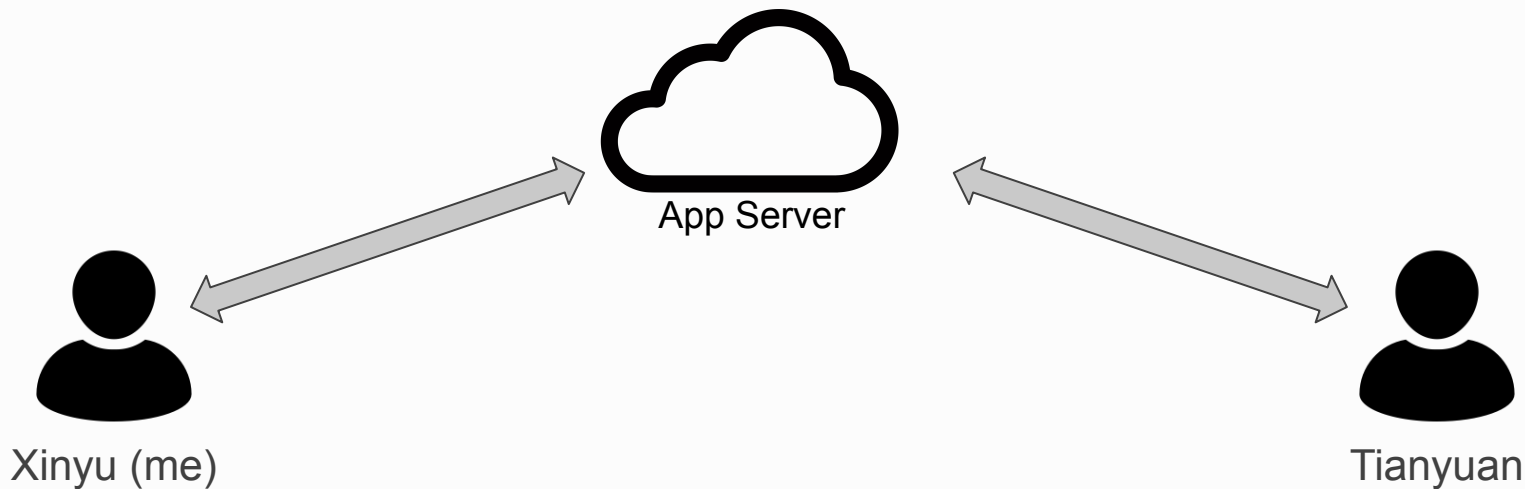
NDN Workspace

Local-first Collaborative Editing over NDN

Tianyuan Yu, **Xinyu Ma**, Varun Patil, et al.

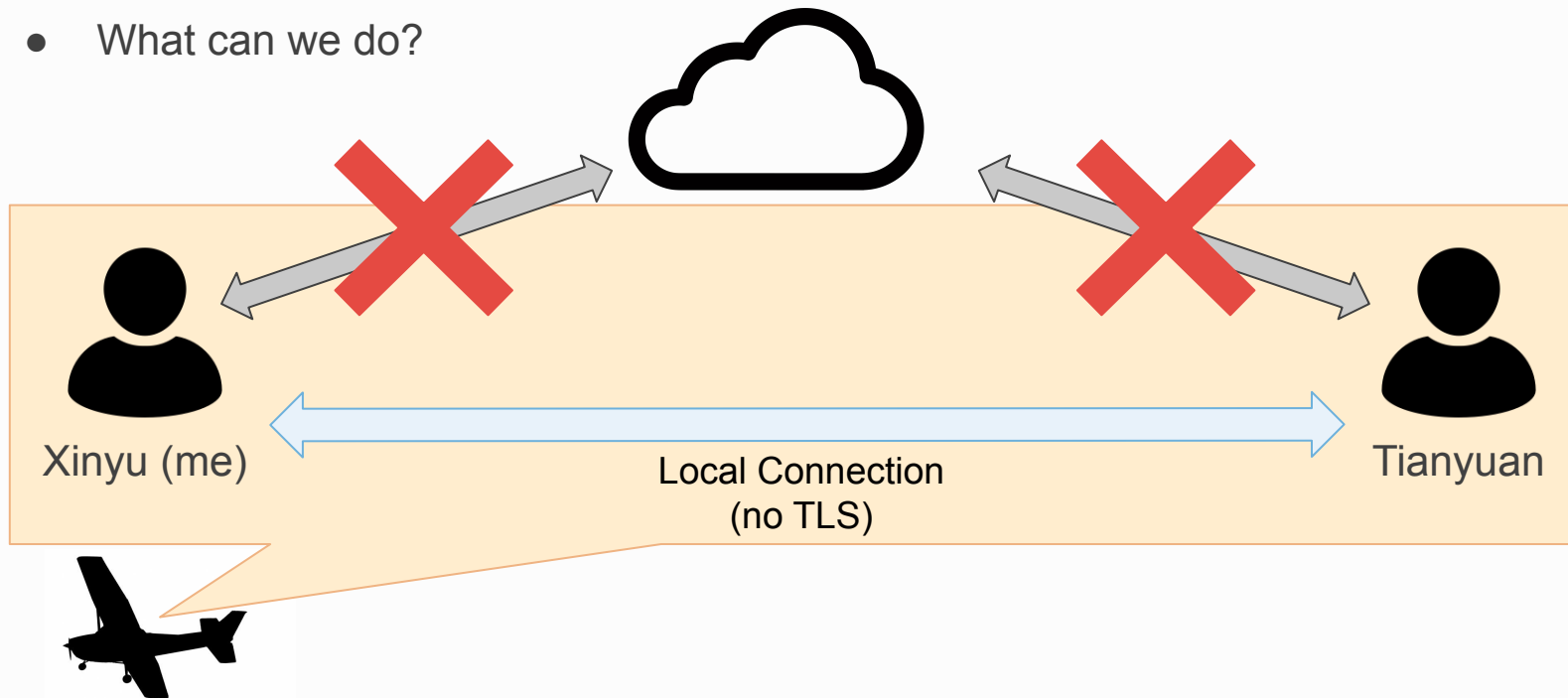
Scenario

- Suppose I and Tianyuan are writing a paper together
- We use collaborative apps to resolve conflicts



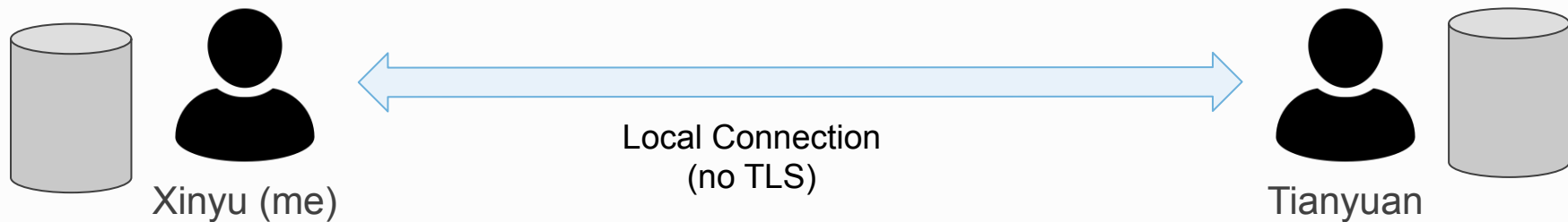
Scenario

- However, when we are both on an airplane and do not have access to the Internet
- What can we do?



Scenario

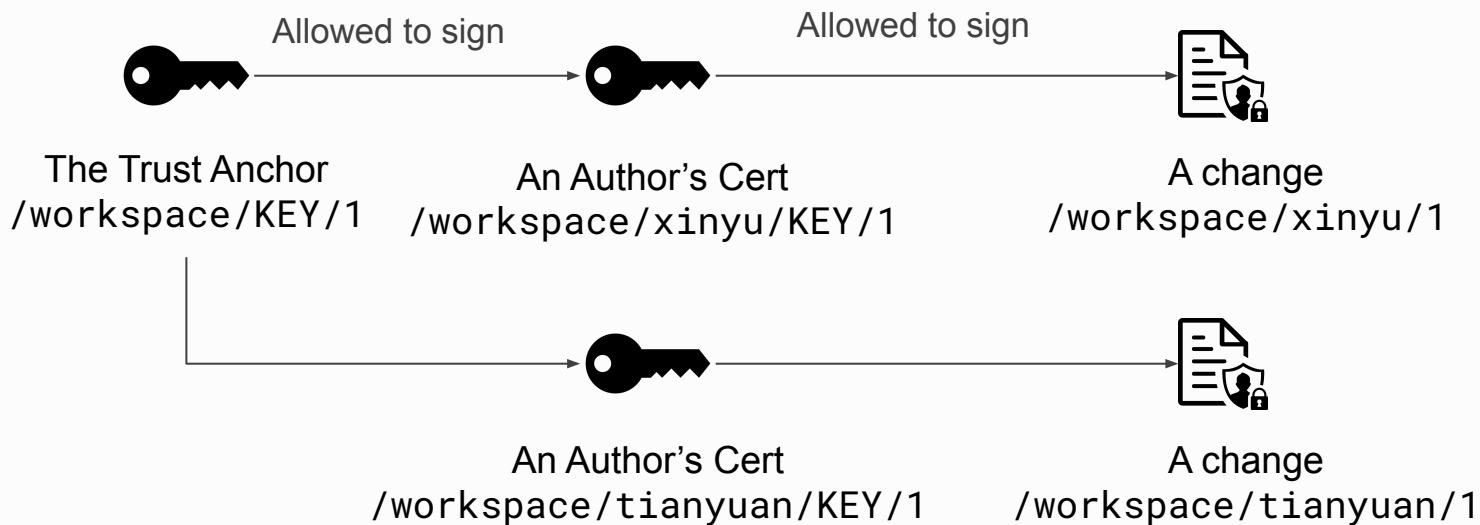
- However, when we are both on an airplane and do not have access to the Internet
- We need a new collaborative editing app that
 - Allows access data offline -> **local first**
 - Supports secure local connection -> Where **NDN** helps!



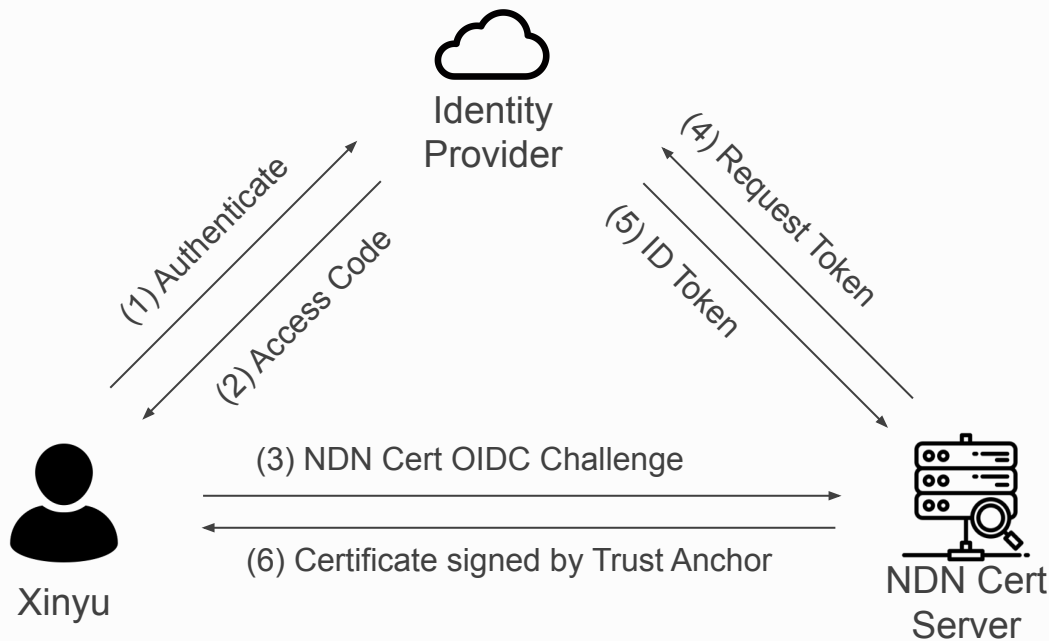
In an open environment, how can a device trust received data?

Direct user-to-user security with no trusted server to verify each party

NDN Trust Zone

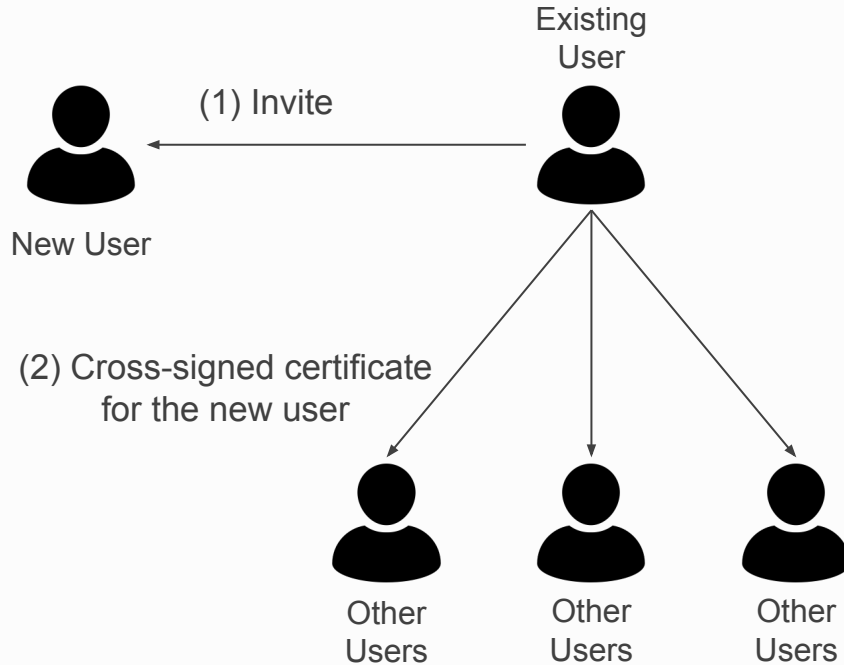


Bootstrapping using OIDC



- The IdP is only used for bootstrapping
- Other existing trust relations can also be used (email, DNS, etc.)
- The application does not depend on any existing service after bootstrapping

Web-of-Trust (work in progress)



How does the Workspace work?

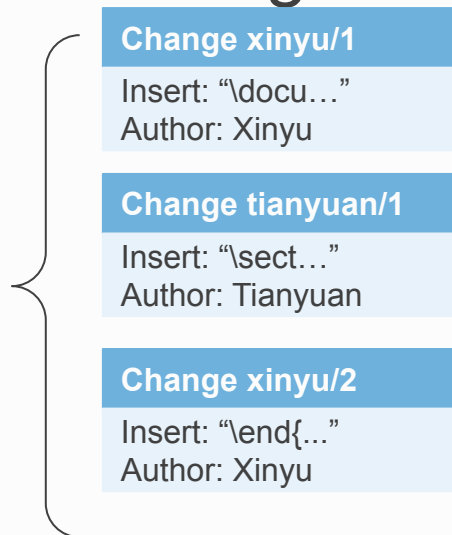
Demonstrate *data-centric communication* by an example

Data Model: Event Sourcing

```
1 \documentclass[12pt]{article}
2 \usepackage{lingmacros}
3 \usepackage{tree-dvips}
4 \begin{document}
5
6 \section*{Section 1}
7
8
9
10 \end{document}
11
```

The Document

- Mutable
- Multiple authors
- Viewed in the frontend



Changes

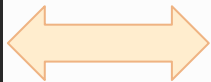
- Immutable
- Single producer
- Persistent in storage
- Propagated in the network

- Fit into NDN's **Data** packet concept
- **Securely** tracking who did what
- **Sync**: when getting same set of changes => same version

Eventual Consistency: CRDT

```
1 \documentclass[12pt]{article}
2 \usepackage{lingmacros}
3 \usepackage{tree-dvips}
4 \begin{document}
5
6 \section*{Section 1}
7
8
9
10 \end{document}
11
```

The Document



CRDT

Change xinyu/1

Insert: "\docu..."
Author: Xinyu

Change tianyuan/1

Insert: "\sect..."
Author: Tianyuan

Change xinyu/2

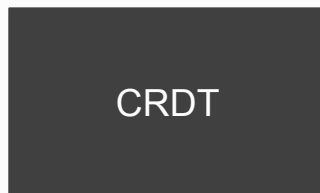
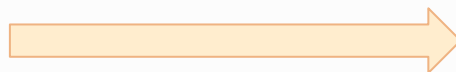
Insert: "\end{..."
Author: Xinyu

- **Sequentialize** changes to solve conflicts
- Mature algorithms and implementations

Changes

When I make a change ...

```
1 \documentclass[12pt]{article}
2 \usepackage{lingmacros}
3 \usepackage{tree-dvips}
4 \begin{document}
5
6 \section*{Section 1}
7
8
9
10 \end{document}
11
```

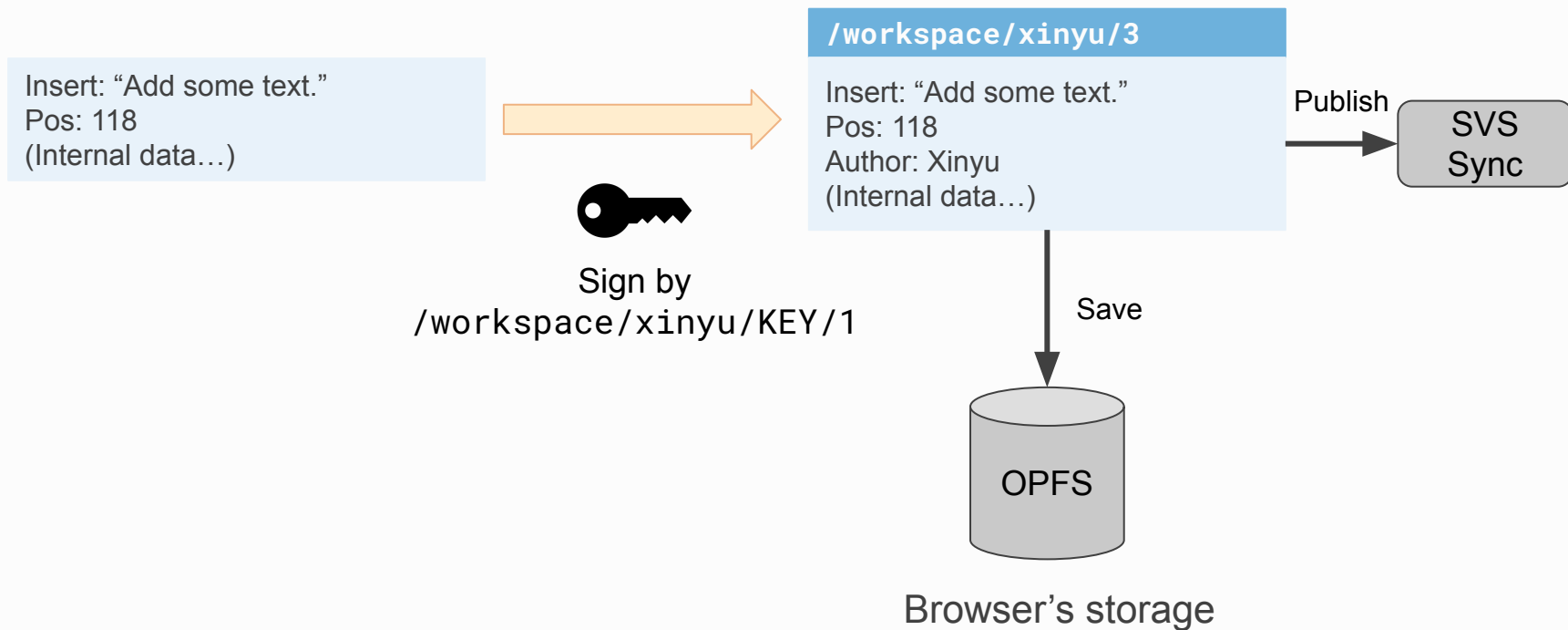


```
1 \documentclass[12pt]{article}
2 \usepackage{lingmacros}
3 \usepackage{tree-dvips}
4 \begin{document}
5
6 \section*{Section 1}
7
8 Add some text.
9
10 \end{document}
11
```

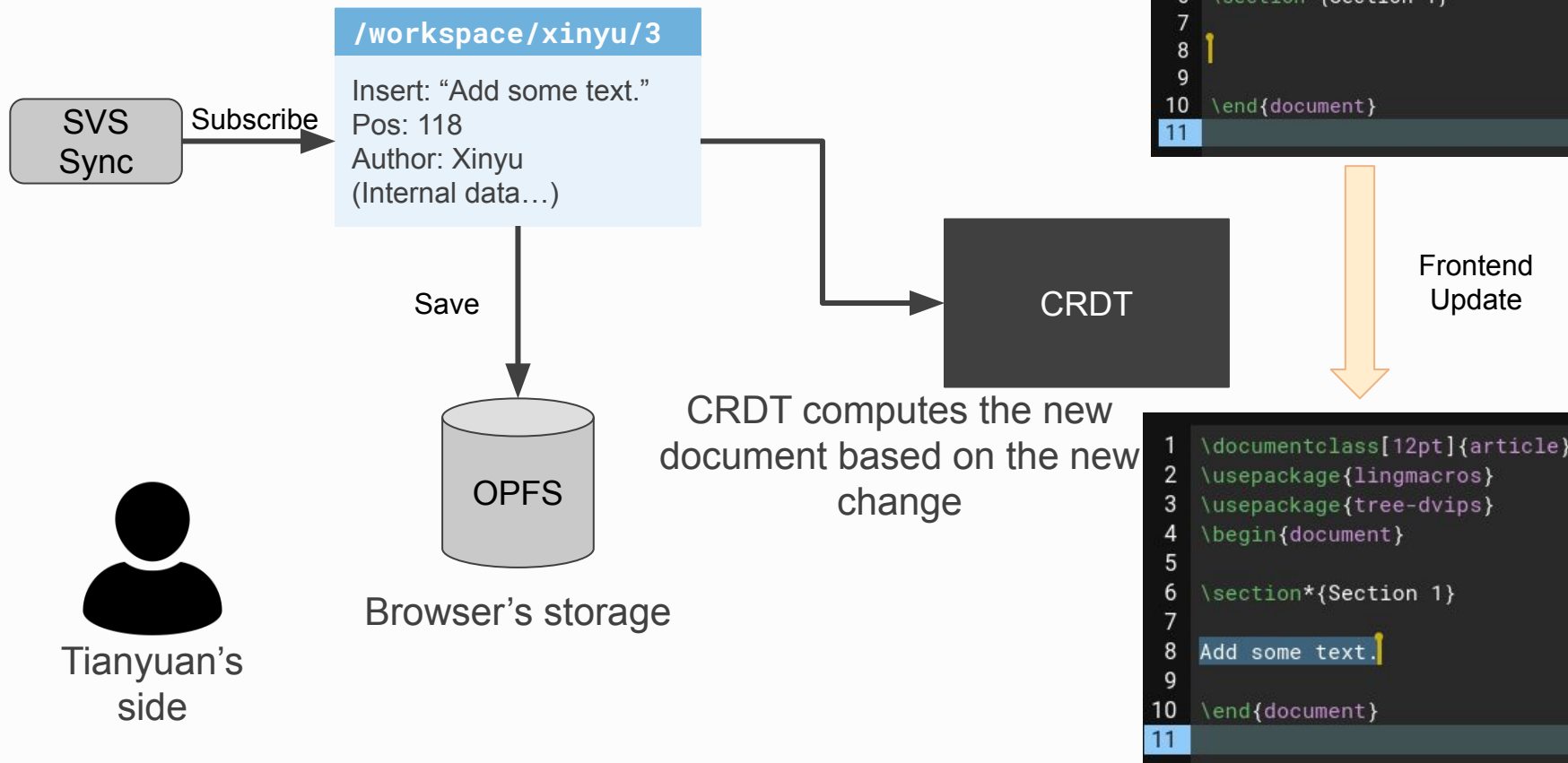
CRDT captures this edit and
outputs a new **change**

Insert: "Add some text."
Pos: 118
(Internal data...)

Wrap it as a Data packet



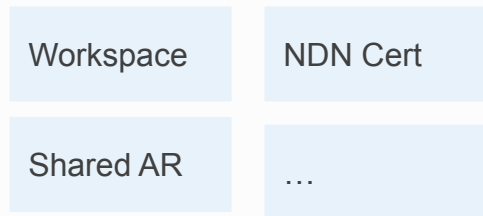
Apply the change



But users may not be online
simultaneously

Asynchronous communication with Repo, a generic
in-network *storage*

NDN Repo



Testbed Repo

- Generality as network service
- Agnostic to application security

/workspace/xinyu/3

Insert: "Add some text."
Pos: 118
Author: Xinyu
(Internal data...)

/Workspace/xinyu



/Workspace/tianyuan



NDN Testbed

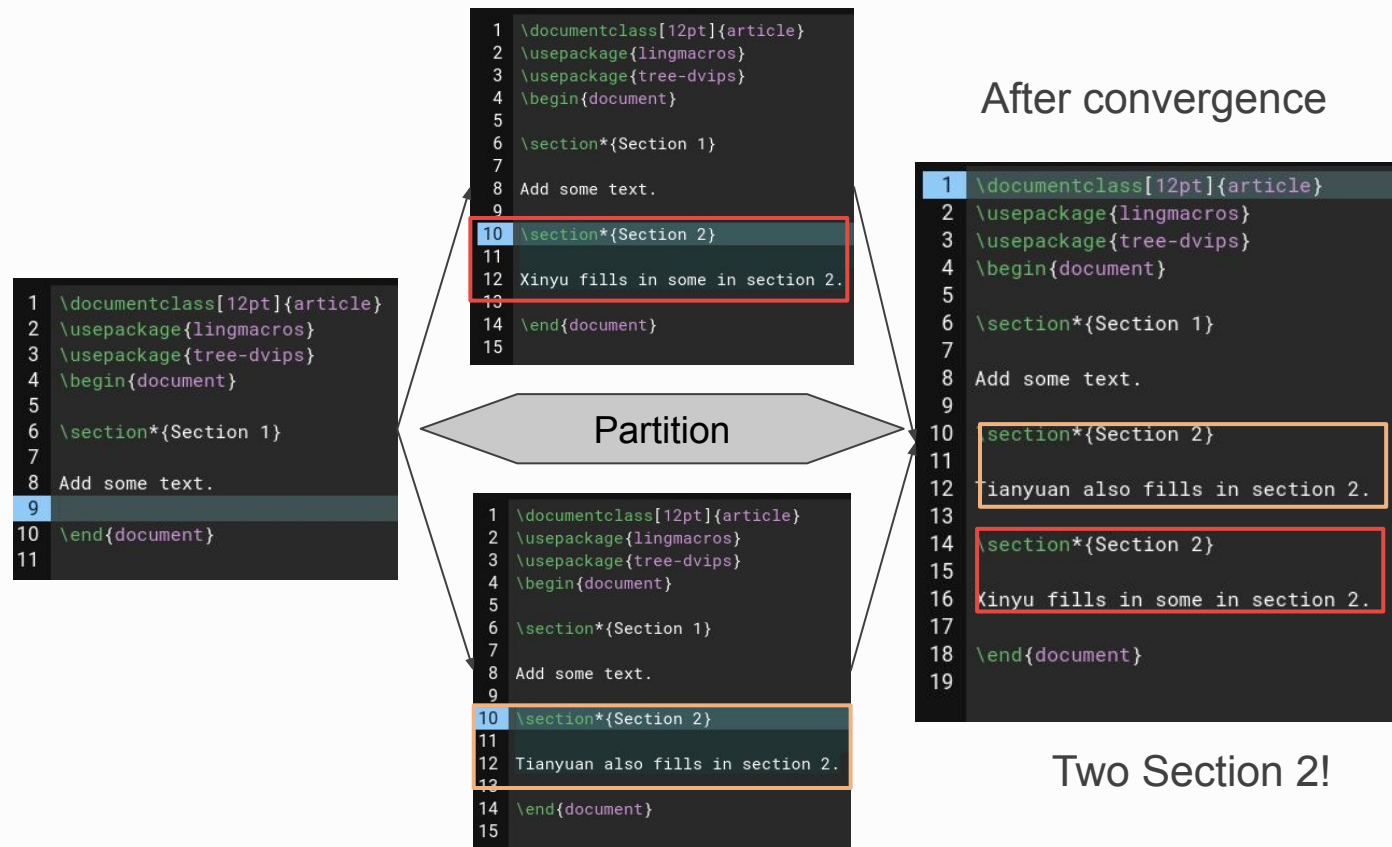


/Workspace/varun

How does it differ from cloud apps?

Local-first Collaborative Editing over NDN

Offline availability comes with limitations ...

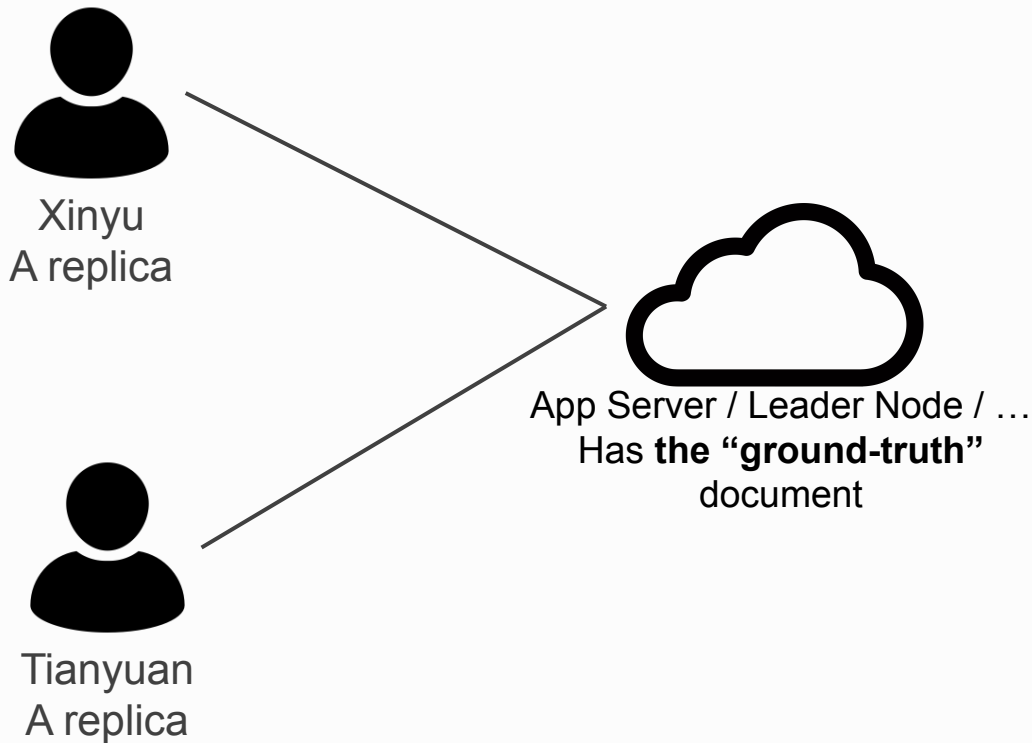


CRDT's consistency

We will eventually see the **same** result.

Not guaranteed to be the **right** result.

How the cloud handles this



Server is the door keeper

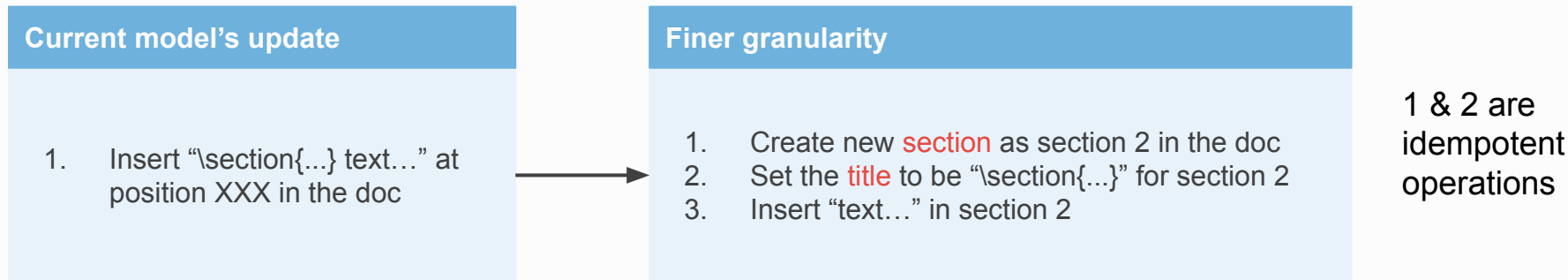
- The server decides the order of requests
- The server validates all requests.
- The server can do semantic check and solve issues.
- The server can reject updates it does not like.

But

- The user has to connect to it
- The user has to trust it

How about our app

- No special door keeper holding the ground truth.
- All legit requests (securely verified) should accepted.
- Consider the following
 - Shared documents more tolerant than bank account. Can fix small issues manually as needed.
 - 2. Can make better use of the algorithm to fix it, finer granularities will help



The trade-off

- Prefer **availability** over “semantic correctness guarantee”
- Works in a wide range of application scenarios:



- Text editing is not special
 - We can generalize it into a **Key-Value JSON document** storage

How does it differ from git?

Local-first Collaborative Editing over **NDN**

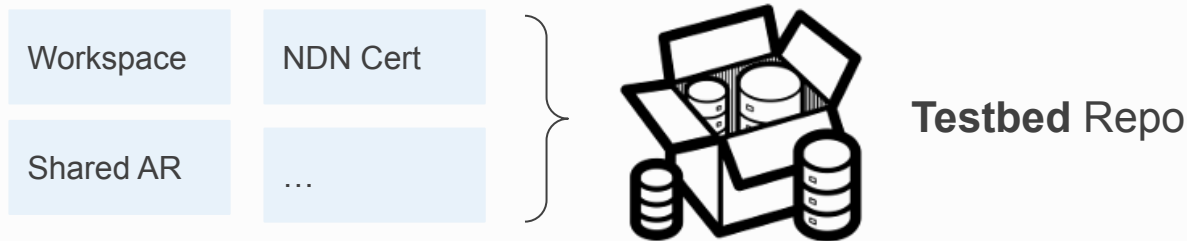
Using NDN as data transport

- Real-time collaboration enabled by NDN Sync
 - No need to manual `git pull`.
 - Other users get notified as you typing
- Systematic and automated security
 - Git can sign commits, but has no built-in design on distribution of the signing keys.
 - In NDN a named certificate can be fetched directly
 - With trust schema defining their relations and usages

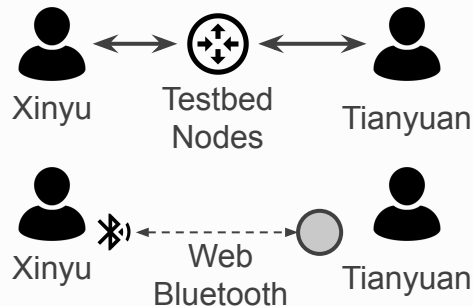
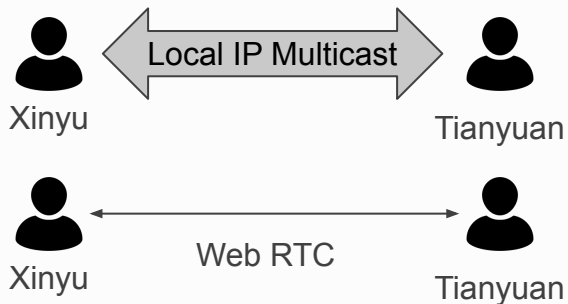


NDN: data-centric networking

- Exploit NDN Repos, a *generic Data storage* as a networking service



- Use *any NDN channel* as rendezvous points, not any specific server.
 - NDN Security is not attached to the channel



Summary

Comparison Table

Services / Software	Phone & Tablet	Offline	Collaboration	Free	Security
Overleaf	✓	✗	✓	⚠	TLS & Login
Git	⚠	✓	⚠ (no real-time)	✓	GPG & SSH
NDN Workspace	✓	✓	✓	✓	Trust Schema

Takeaways

- NDN enables secure communication between any parties, independent of the channel
- Fundamental building blocks of NDN applications: **Security, Sync, Storage**
 - Semantically named data objects allow generic network services
- NDN + CRDT solves real-world problems with offline availability
 - Trade-offs with local first concept

Thanks

Workspace URL: <https://ndn-workspace.web.app/>

Contributions are welcome

