To be continued....

There are a number of future directions to explore:

To be continued....

There are a number of future directions to explore:

Identify weaker conditions for well-formedness

To be continued....

There are a number of future directions to explore:

Identify weaker conditions for well-formedness

"Efficiency"

To be continued....

There are a number of future directions to explore:

Identify weaker conditions for well-formedness

"Efficiency"

Subscriptions are hard to determine

To be continued....

There are a number of future directions to explore:

- Identify weaker conditions for well-formedness
- "Efficiency"
- Subscriptions are hard to determine
- Relax some of our assumptions

To be continued....

There are a number of future directions to explore:

- Identify weaker conditions for well-formedness
- "Efficiency"
- Subscriptions are hard to determine
- Relax some of our assumptions
  - Compensations
  - Unreliable propagation

To be continued....

There are a number of future directions to explore:

- Identify weaker conditions for well-formedness
- "Efficiency"
- Subscriptions are hard to determine
- Relax some of our assumptions
  - Compensations
  - Unreliable propagation
  - Adversarial contexts

To be continued....

There are a number of future directions to explore:

- Identify weaker conditions for well-formedness
- "Efficiency"
- Subscriptions are hard to determine
- Relax some of our assumptions
  - Compensations
  - Unreliable propagation
  - Adversarial contexts
- .....

Thank you!

Summary

An interesting paradigm grounded on principles for local-first software

Summary

An interesting paradigm grounded on principles for local-first software

We defined an operational semantics that captures the platform of Actyx AG

Summary

An interesting paradigm grounded on principles for local-first software

We defined an operational semantics that captures the platform of Actyx AG

We introduced behavioural types to specify and verify eventual consistency

Summary

An interesting paradigm grounded on principles for local-first software

We defined an operational semantics that captures the platform of Actyx AG

We introduced behavioural types to specify and verify eventual consistency

The key idea is to trade consistency for availability: temporary inconsistency are tolerated provided that they can be resolved at some point