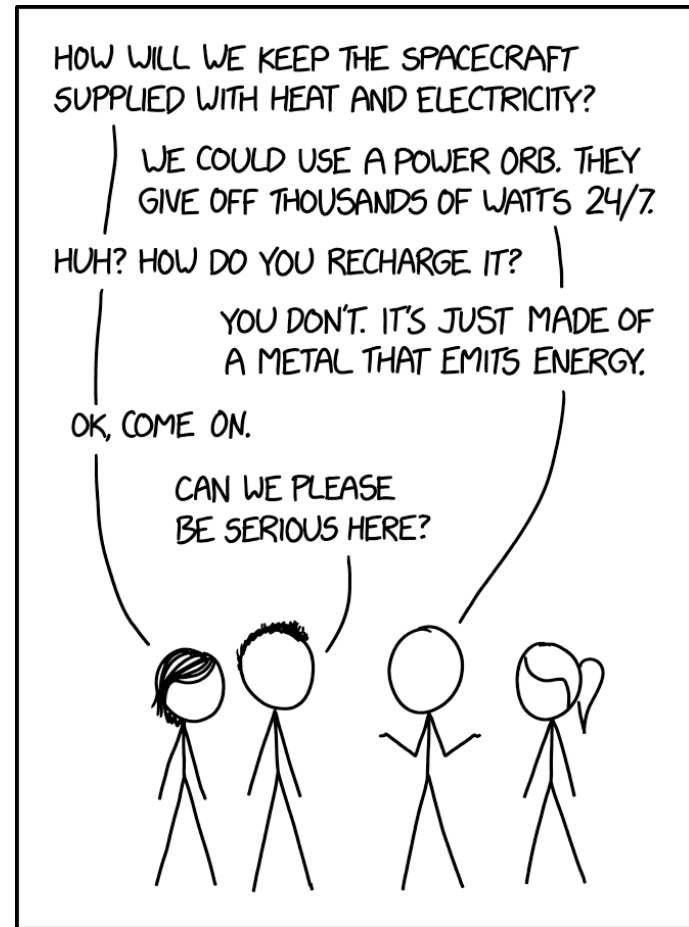


# Architecting Ag Data Dreams

## Bounded Context in Time and Space

Aaron Ault  
OATS Center  
at Purdue University



# Interoperability



Farmer

"I don't **have to spend time** messing with anything."

# Interoperability



Farmer

"I don't **have to spend time** messing with anything."



Industry

"We all use the **same data model**."

# Interoperability



Farmer

"I don't **have to spend time** messing with anything."

MINUS



Industry

"We all use the **same data model**."

EQUALS

---

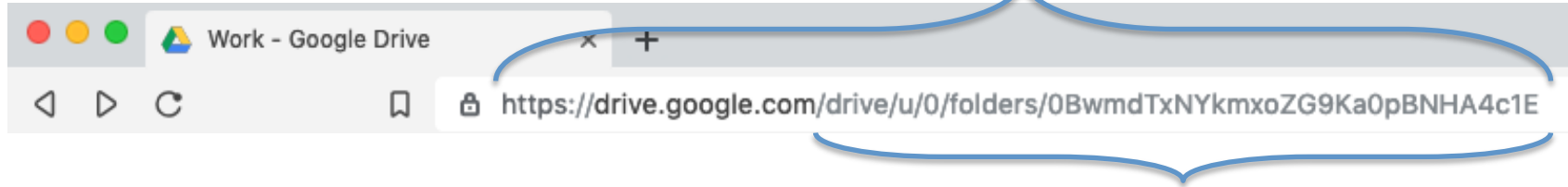
## API Connections

# OADA/Trellis API Connections

Concepts for Industry-Wide Data Sharing API's

# OADA/Trellis API Connections

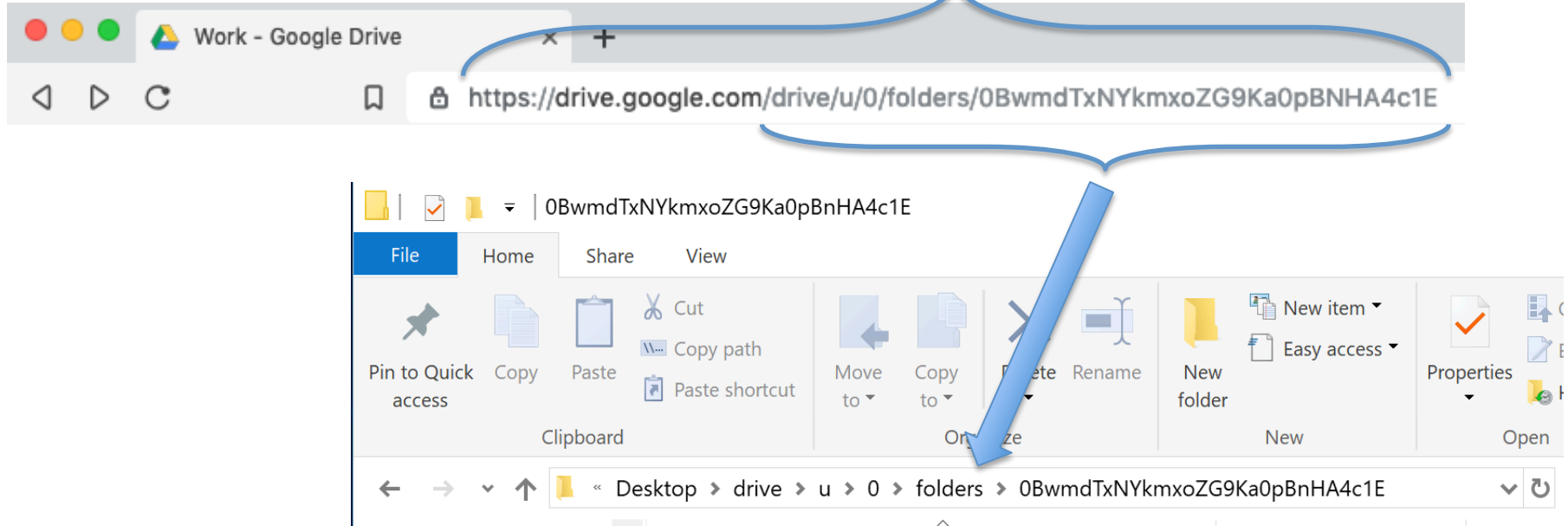
API's identify Resources with URL's



What does this part look like?

# OADA/Trellis API Connections

API's identify Resources with URL's



Filesystem!

# OADA/Trellis API Connections

API's Identify Resources with URL's

OADA URL's use Links to form a graph  
(like a filesystem + symbolic links)



# OADA/Trellis API Connections

Now imagine if an agronomist had 3 different farmer clients share their Google Drive folders with him, and they all use the same folder structure and filenames

sources with URL's

links to form a graph



Share

Name	Owner	Last modified
fields	me	3:03 PM me
harvest	me	2:52 PM me
livestock	me	2:53 PM me
planting	me	2:52 PM me
spraying	me	2:52 PM me

His software could then pull whatever it needed for him automatically, and save back his recommendations automatically

# OADA/Trellis API Connections

API's identify Resources with URL's

OADA URL's use Links to form a graph

Graph structure has many roles:

Semantic: organizing content by type,

Sharing: organizing content by privacy,

Indexing: organizing content for querying,

Streaming/Updating/Versioning/Concurrency/Caching (the hard stuff)

# Role 1: Semantic

/fields  
/machines  
/certifications  
/spraying  
/harvest  
/harvest/as-harvested/yield-maps  
/...

Defines "Categories" of data

# Role 2: Sharing

/fields  
/machines  
/certifications  
/spraying  
/harvest  
/harvest/as-harvested/yield-maps/**years/2018**  
/...

Group items into trees (folders) to share with others.  
Like here, this allows sharing only 2018's yield maps.

Remember the Google Drive example earlier?

# Role 2: Sharing

/fields  
/machines  
/certifications  
/spraying  
/harvest  
/harvest/as-harvested/yield-maps/years/2018  
/...

Problem:  
You shared this node




But recipient lost this  
semantic structure

Group items into trees (folders) to share with others.  
Need Content Types and Traveling Context  
Sharing a Filtered Graph is sometimes a better fit.

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

If there are 13 rows in this data set,  
no problem. Just put them in one  
resource because the client can  
always pull them all.



id	time	lat	lon	mass (bu)	moisture (%)	area (ac)
1	1550957130	41.06496	-86.216011	14	19.7	0.07
2	1550957131	41.06596	-86.214011	14.3	19.4	0.07
3	1550957132	41.06696	-86.212011	14.1	19.9	0.07
4	1550957133	41.06796	-86.210011	15.8	19.4	0.07
5	1550957134	41.06896	-86.208011	15.8	19.7	0.07
6	1550957135	41.06996	-86.206011	18.2	19.7	0.07
7	1550957136	41.07096	-86.204011	14.7	19.4	0.07
8	1550957137	41.07196	-86.202011	14	19.9	0.07
9	1550957138	41.07296	-86.200011	14	19.9	0.07
10	1550957139	41.07396	-86.198011	16.1	19.7	0.07
11	1550957140	41.07496	-86.196011	16.1	19.4	0.07
12	1550957141	41.07596	-86.194011	14.7	19.7	0.07
13	1550957142	41.07696	-86.192011	14	19.9	0.07

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

130M rows, problem!

Databases FTW!

id	time	lat	lon	mass (bu)	moisture (%)	area (ac)
1	1550957130	41.06496	-86.216011	14	19.7	0.07
2	1550957131	41.06596	-86.214011	14.3	19.4	0.07
3	1550957132	41.06696	-86.212011	14.1	19.9	0.07
4	1550957133	41.06796	-86.210011	15.8	19.4	0.07
5	1550957134	41.06896	-86.208011	15.8	19.7	0.07

...

135654013	1550957140	41.07496	-86.196011	16.1	19.4	0.07
135654014	1550957141	41.07596	-86.194011	14.7	19.7	0.07
135654015	1550957142	41.07696	-86.192011	14	19.9	0.07

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

A database builds an Index to find some things  
without needing to look at everything

id	time	lat	lon	mass (bu)	moisture (%)	area (ac)	lat	id
1	1550957130	41.06496	-86.216011	14	19.7	0.07	41.06496	10
2	1550957131	41.06596	-86.214011	14.3	19.4	0.07	41.06596	2
3	1550957132	41.06696	-86.212011	14.1	19.9	0.07	41.06796	4
4	1550957133	41.06796	-86.210011	15.8	19.4	0.07	41.06896	5
5	1550957134	41.06896	-86.208011	15.8	19.7	0.07	41.06991	11
...							41.06996	6
							41.07096	1
							41.07196	8
							41.07296	9
							41.07496	12
							41.07596	7
135654013	1550957140	41.07496	-86.196011	16.1	19.4	0.07	41.07694	13
135654014	1550957141	41.07596	-86.194011	14.7	19.7	0.07	41.07696	3
135654015	1550957142	41.07696	-86.192011	14	19.9	0.07		

Dataset

Latitude Index

If index is sorted,  
can find ranges



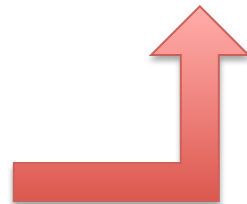
# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

A database builds an Index to find some things  
without needing to look at everything

/harvest/as-harvested/yield-maps/years/2018/points.csv?lat=40.0798&lon=-86.2913

API's sometimes handle these indexes  
with special query parameters on the  
end of the URL like this



# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

A database builds an Index to find some things  
without needing to look at everything

/harvest/as-harvested/yield-maps/years/2018/points.csv?lat=40.0798&lon=-86.2913



/harvest/as-harvested/yield-maps/years/2018/lat/40.079/lon/-86.2913/points.csv

But this is really the same as just making it part of the graph structure

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/points.csv

A database builds an Index to find some things  
without needing to look at everything

/harvest/as-harvested/yield-maps/years/2018/points.csv?lat=40.0798&lon=-86.2913

/harvest/as-harvested/yield-maps/years/2018/lat/40.079/lon/-86.2913/points.csv



Turns out this was an index, too  
(limits the dataset to 1 particular year)

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/fields/Back40/points.csv

id	time	lat	lon	mass (bu)	moisture (%)	area (ac)
1	1550957130	41.06496	-86.216011	14	19.7	0.07
2	1550957131	41.06596	-86.214011	14.3	19.4	0.07
3	1550957132	41.06696	-86.212011	14.1	19.9	0.07
4	1550957133	41.06796	-86.210011	15.8	19.4	0.07
5	1550957134	41.06896	-86.208011	15.8	19.7	0.07

/harvest/as-harvested/yield-maps/years/2018/fields/Front40/points.csv

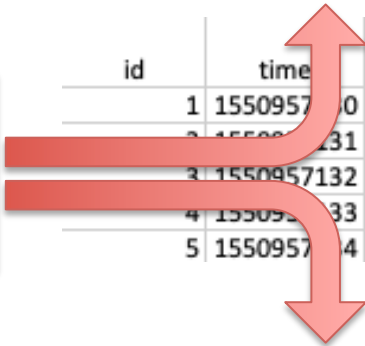
id	time	lat	lon	mass (bu)	moisture (%)	area (ac)
1	1550957130	41.06496	-86.216011	14	19.7	0.07
2	1550957131	41.06596	-86.214011	14.3	19.4	0.07
3	1550957132	41.06696	-86.212011	14.1	19.9	0.07
4	1550957133	41.06796	-86.210011	15.8	19.4	0.07
5	1550957134	41.06896	-86.208011	15.8	19.7	0.07

You have now Graph Indexed a large dataset into smaller "buckets"

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/fields/Back40/points.csv

You have committed a  
Cardinal Sin  
and broke everything



	id	time	lat	lon	mass (bu)	moisture (%)	area (ac)
	1	1550957130	41.06496	-86.216011	14	19.7	0.07
	2	1550957131	41.06596	-86.214011	14.3	19.4	0.07
	3	1550957132	41.06696	-86.212011	14.1	19.9	0.07
	4	1550957133	41.06796	-86.210011	15.8	19.4	0.07
	5	1550957134	41.06896	-86.208011	15.8	19.7	0.07

/harvest/as-harvested/yield-maps/years/2018/fields/Front40/points.csv

These field names are arbitrary, non-unique, and mutable.  
Yet the Ag software industry has chosen them as their core  
indexing, causing no end of headaches for farmers trying to  
manage their data over time.

	lon	mass (bu)	moisture (%)	area (ac)
96	-86.216011	14	19.7	0.07
96	-86.214011	14.3	19.4	0.07
96	-86.212011	14.1	19.9	0.07
96	-86.210011	15.8	19.4	0.07
96	-86.208011	15.8	19.7	0.07

There will come a judgment day...

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/fields/Back40/points.csv

You always get 1 index for free. (You have to store the data somewhere.)

*Make it count.*

*Make it immutable.*

*Make it non-ambiguous.*

# Role 3: Indexing

/harvest/as-harvested/yield-maps/years/2018/fields/Back40/points.csv

You always get 1 index for free.

*Make it count.*

*Make it immutable.*

*Make it non-ambiguous*

## GOGT Data!

Graph-structured

OADA-delivered

Geohash-indexed

Time-indexed

# Role 3: Indexing

/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

Time-indexed

Geohash-indexed  
(7-character =  $\approx 2.5$  ac/tile in Midwest US)



# Role 3: Indexing

/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

Time-indexed

Geohash-indexed  
(7-character =  $\approx 2.5$  ac/tile in Midwest US)

Geohash: a way to "tile" GPS points

Lon	Lat
-86.216011	41.070960

-8461.201760091610

Geohash-like string  
(real geohashes interleave binary)

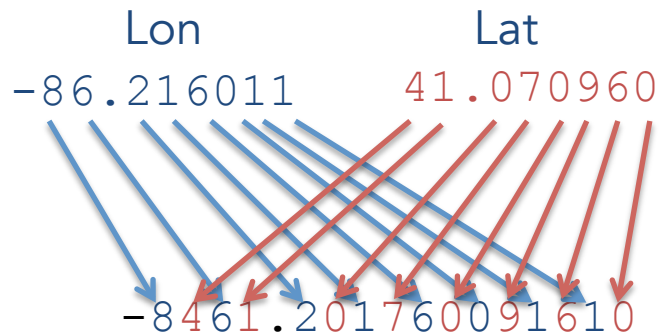
# Role 3: Indexing

/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

Time-indexed

Geohash-indexed  
(7-character =  $\approx 2.5$  ac/tile in Midwest US)

Geohash: a way to "tile" GPS points



Geohash-like string  
(real geohashes interleave binary)

Some Nice Properties for Indexing:

2D => 1D

$\approx$ Immutable (on non-geologic timescales)

Unambiguous

Purely Functional

Fewer bits = Bigger Tiles

Close in "number"  $\approx$  Close in "space"

# Role 3: Indexing

/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

GOGT is a great option for a lot of Ag-related Spatial Data:

As-planted  
As-harvested  
As-applied  
Soil tests  
LiDAR Elevations  
Weather Stations  
...etc...

(Midwest US)

Indexing:

Geohash:

Lon

-86.21

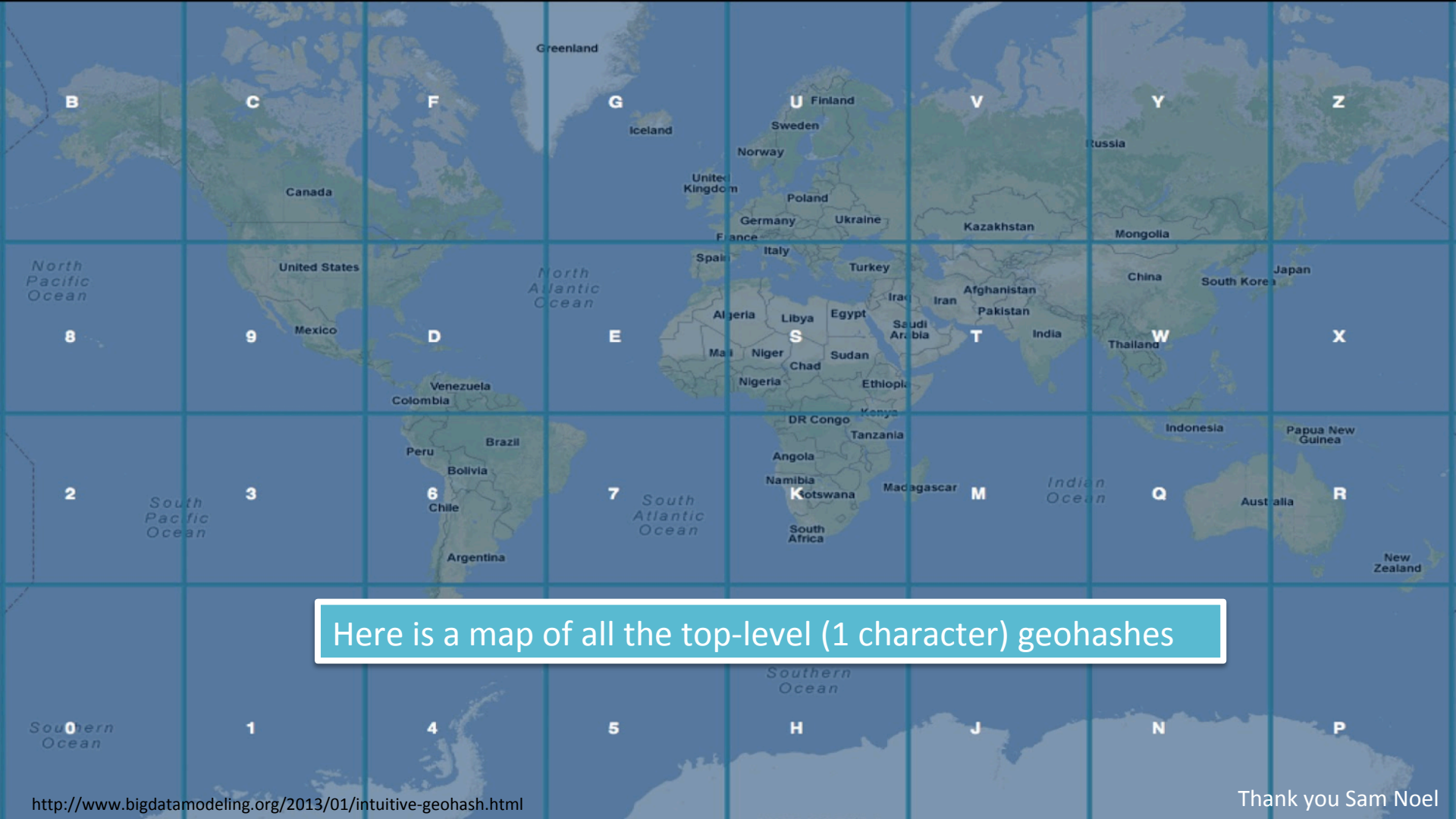
-8461.201760091610

Geohash-like string  
(real ones interleave binary)

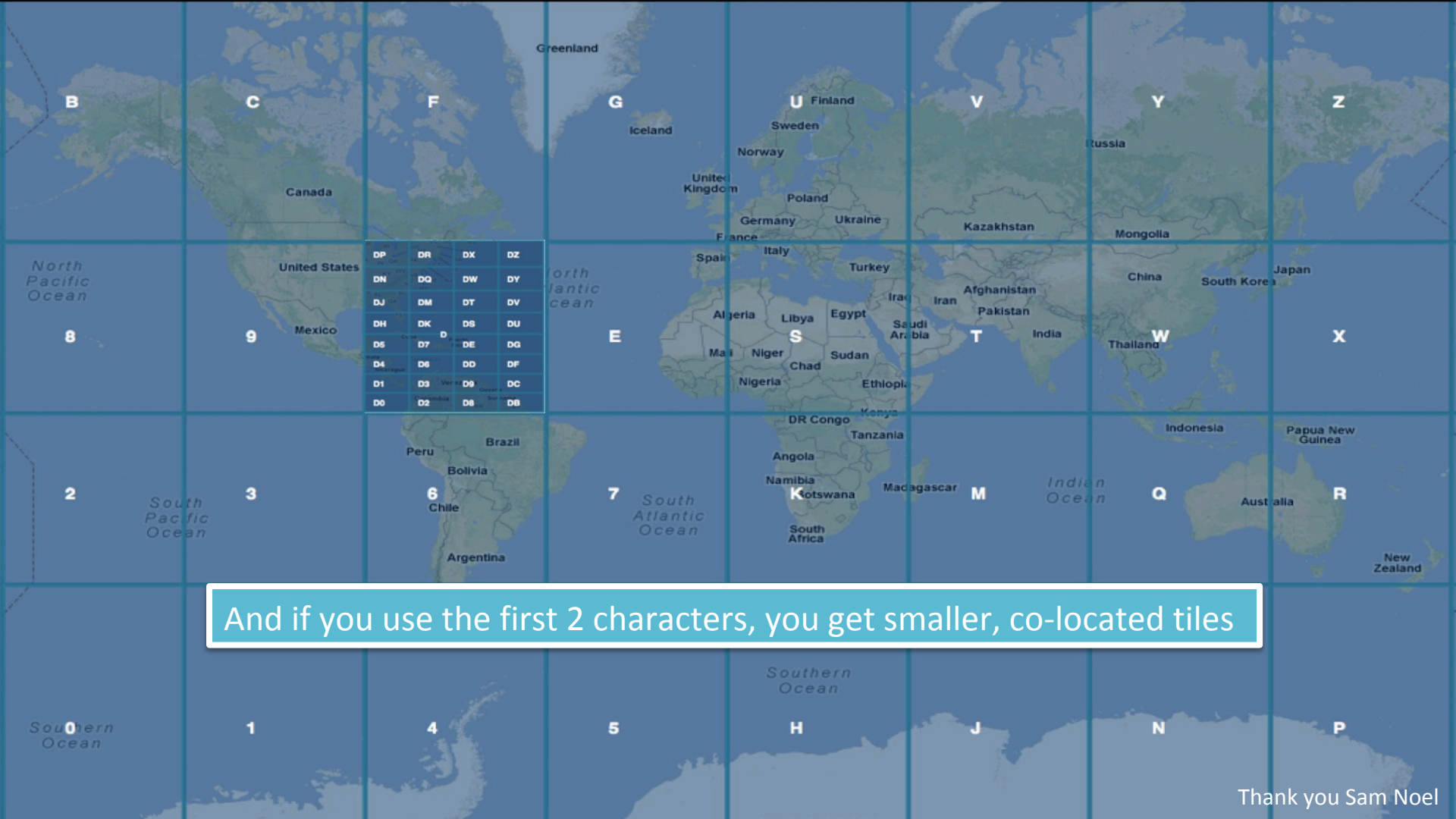
Purely Functional

Fewer bits = Bigger Tiles

Close in "number"  $\approx$  Close in "space"



Here is a map of all the top-level (1 character) geohashes



And if you use the first 2 characters, you get smaller, co-located tiles

DP68RPVN7



DP	DR	DX	DZ
DN	DQ	DW	DY
DJ	DM	DT	DV
DH	DK	DS	DU
DS	DT	DE	DG
D4	D6	DD	DF
D1	D3	D9	DC
D0	D2	D8	DB

This 9-character tile is therefore in the "DP" larger tile.



# Role 3: Indexing

/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

Dynamic Index: build deeper trees as data grows

solves pagination problems

allows lazy-loaded complexity

Downside: shifts much burden to clients, but can scale infinitely even in edge cases

Dynamic Alphabetical Index (for arbitrary strings):

DynamoDB-style hash ring

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching

MachineHealth.com

Benchmarking  
Service

Recommendations  
Service

Field Sensors

Farmer's Cloud of  
Choice

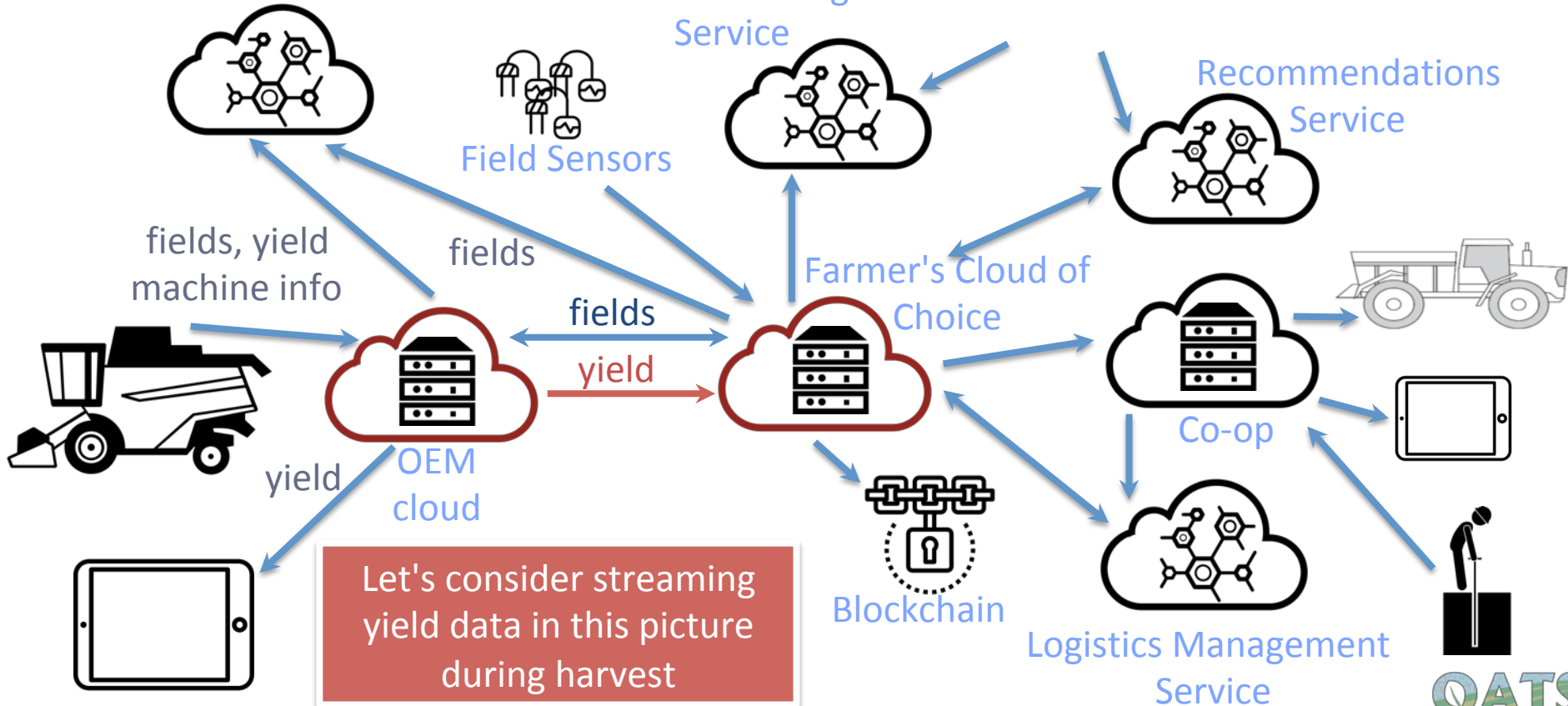
Co-op

Blockchain

Logistics Management  
Service

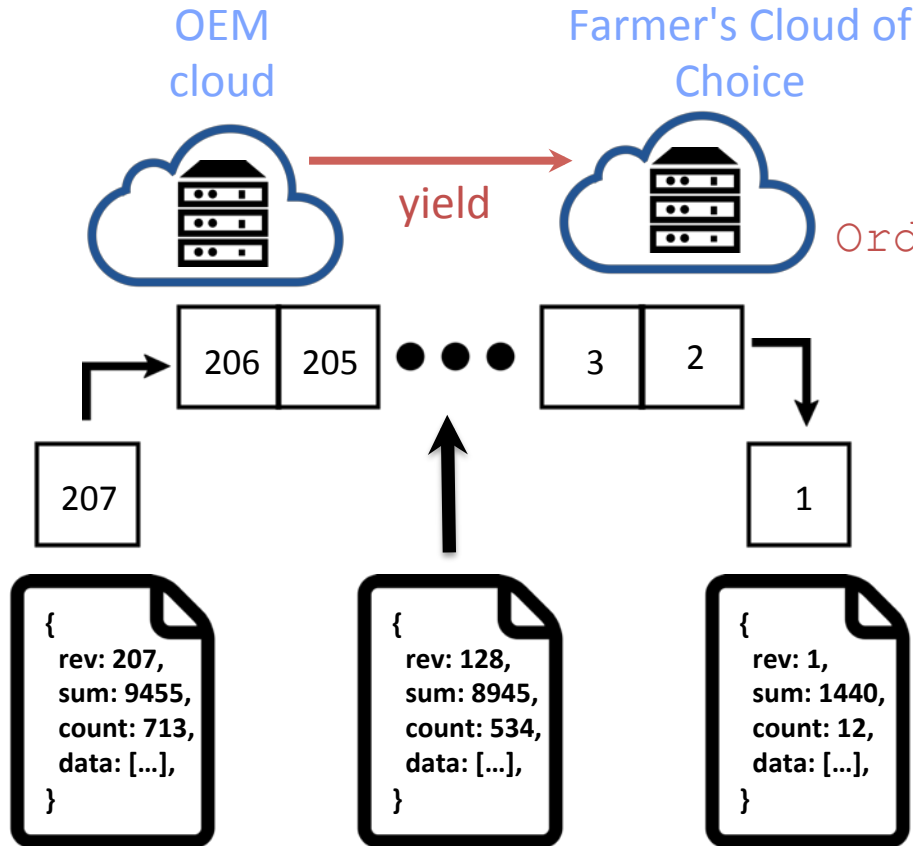
Let's consider streaming  
yield data in this picture  
during harvest

OATS





# Role 4: Streaming/Updating/Versioning/Concurrency/Caching



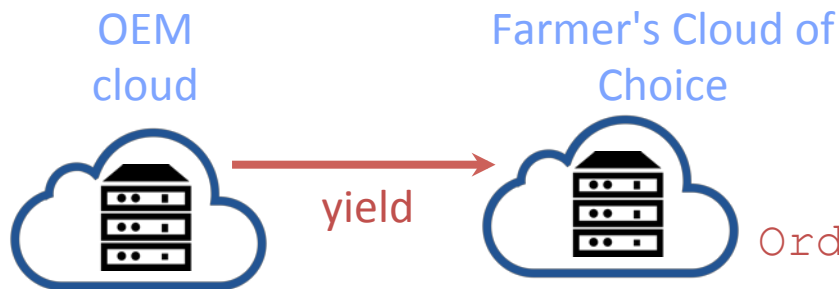
After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

Ordered Stream of Idempotent Merges

Note: merge of UUID's is a  
Convergent replicated data type (CRDT)

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching

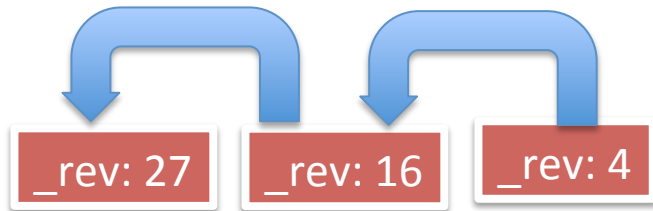


After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

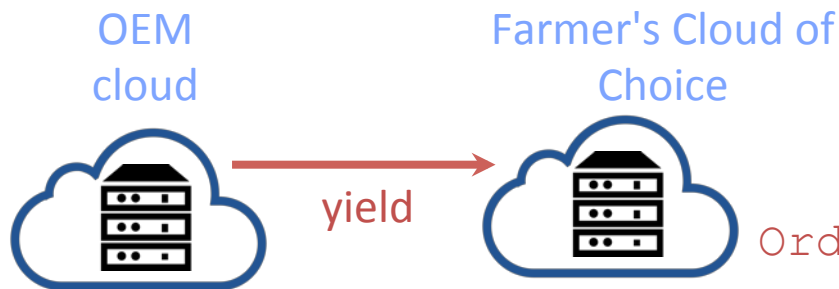
Ordered Stream of Idempotent Merges

Changes "bubble up" Versioned Links



/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching



After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

Ordered Stream of Idempotent Merges

Changes "bubble up" Versioned Links

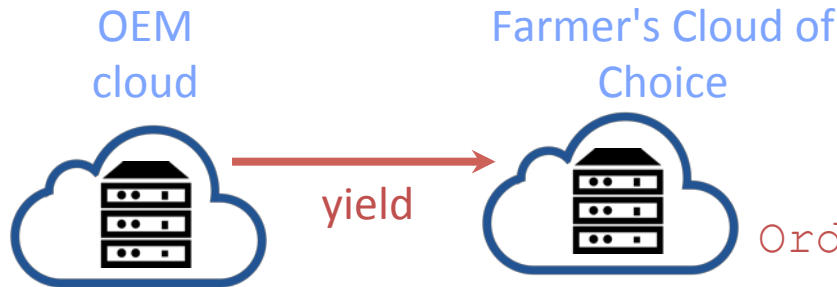
Watch a node in a graph for changes

Replay OSIM stream at remote



</harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv>

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching



After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

Ordered Stream of Idempotent Merges

Changes "bubble up" Versioned Links

Watch a node in a graph for changes

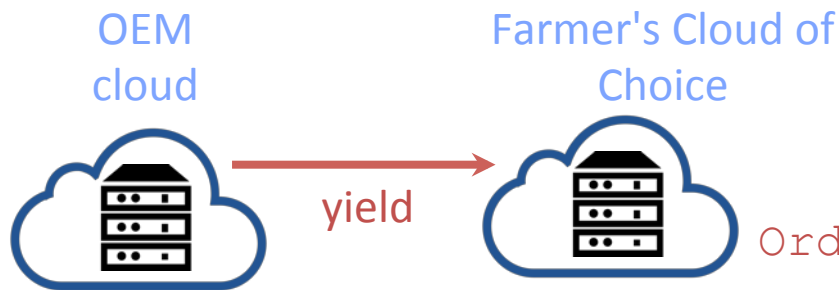
Replay OSIM stream at remote

`_rev`'s for Optimistic Locking

"Only write this data if `_rev` is still 4"

`/harvest/as-harvested/yield-maps/year-index/2018/geohash-index/dp69whe/points.csv`

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching



After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

Ordered Stream of Idempotent Merges

Changes "bubble up" Versioned Links

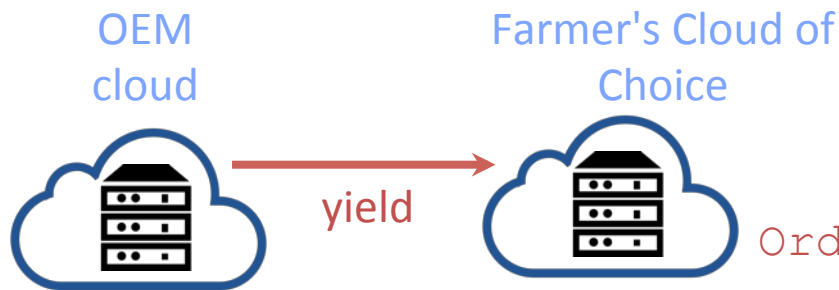
Watch a node in a graph for changes

Replay OSIM stream at remote

\_rev's for Optimistic Locking

oada-cache to cache in-browser or on-server

# Role 4: Streaming/Updating/Versioning/Concurrency/Caching



After initial copy,  
how to keep in sync?

All writes reduce to OSIM:

Ordered Stream of Idempotent Merges

Changes "bubble up" Versioned Links

Watch a node in a graph for changes

Replay OSIM stream at remote

\_rev's for Optimistic Locking

oada-cache to cache in-browser or on-server

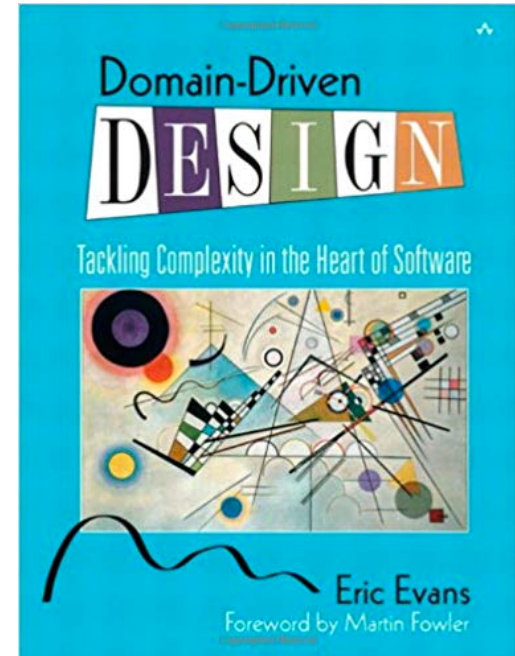
Coming soon: Forward+Reverse OSIM!

We have too many requirements  
for one graph structure to solve them all

# We have too many requirements for one graph structure to solve them all

"Total unification of the domain model for a large system will not be feasible or cost-effective"

– Eric Evans, Domain Driven Design, p332, circa 2004





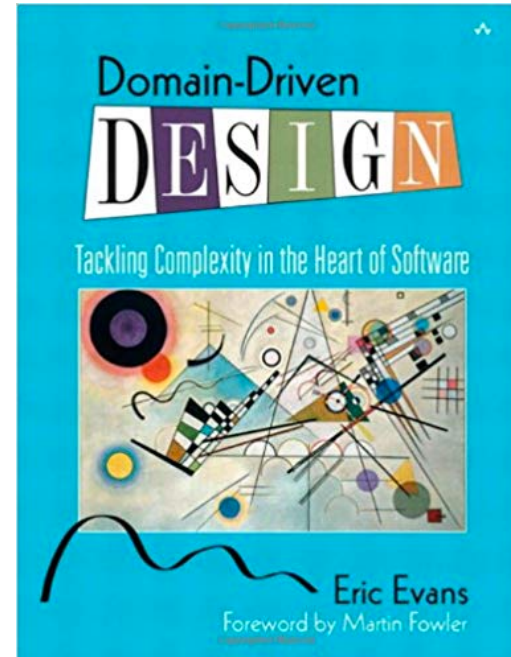
# We have too many requirements for one graph structure to solve them all

"Total unification of the domain model for a large system will not be feasible or cost-effective"

– Eric Evans, Domain Driven Design, p332, circa 2004

and that's ok...

-- Aaron Ault, OATS Conf. 2019



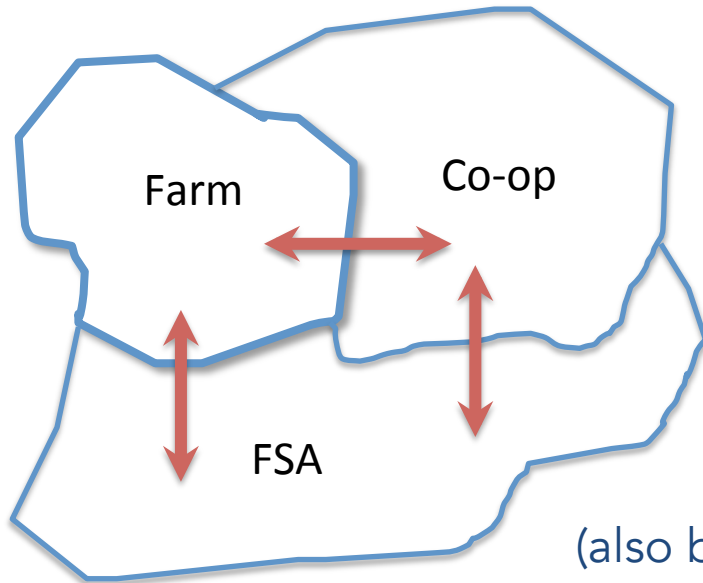
# Dealing With Complexity: DDD

## Bounded Context

# Dealing With Complexity: DDD

Translation is  
likely across boundaries

## Bounded Context



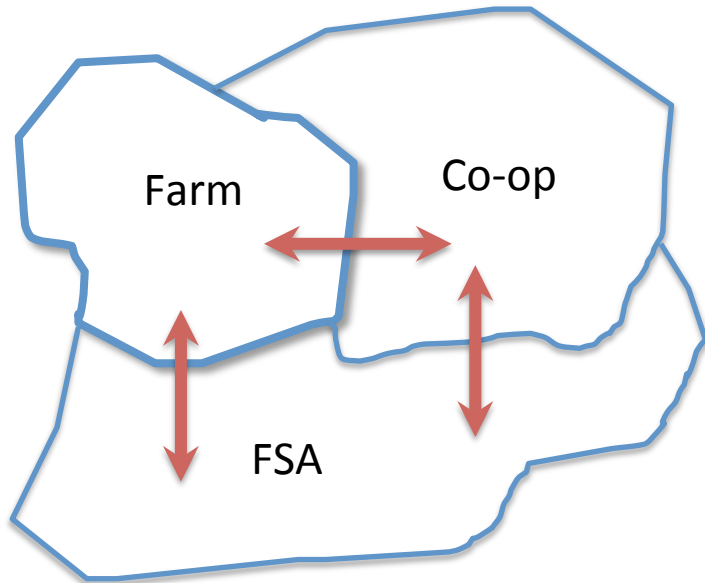
Example: Grower Farm Field

(also bounded contexts likely inside each bubble)

# Dealing With Complexity: DDD

Translation is  
likely across boundaries

## Bounded Context

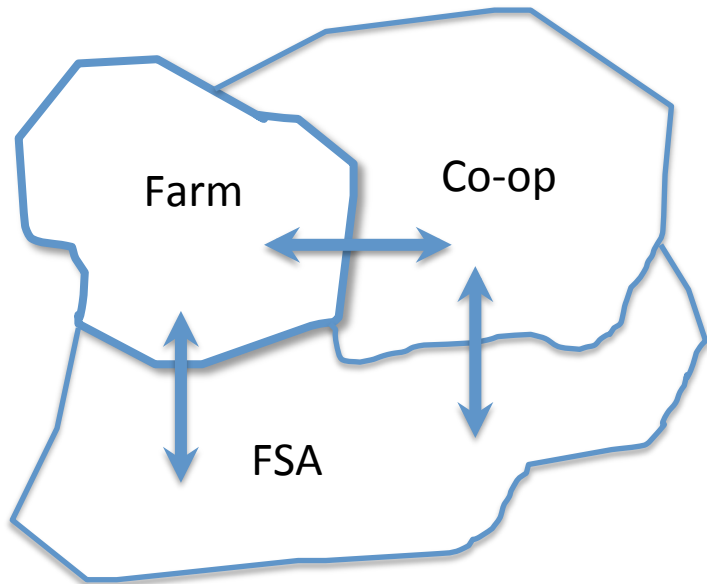


$$Pr(success) \propto \frac{1}{area}$$

You can't build it  
if you can't bound it

# Dealing With Complexity: DDD

## Bounded Context



## Aggregates

Field  
Order  
Note  
Certification  
Operation  
Machine  
Organization  
Operator  
...

# Dealing With Complexity: Microservices

Monoliths are the enemy -- Circa 2012

It's ok to have copies! Use the tool that fits

# Dealing With Complexity: Microservices

Monoliths are the enemy -- Circa 2012

It's ok to have copies! Use the tool that fits

Database per service

Pub/Sub Event Queues

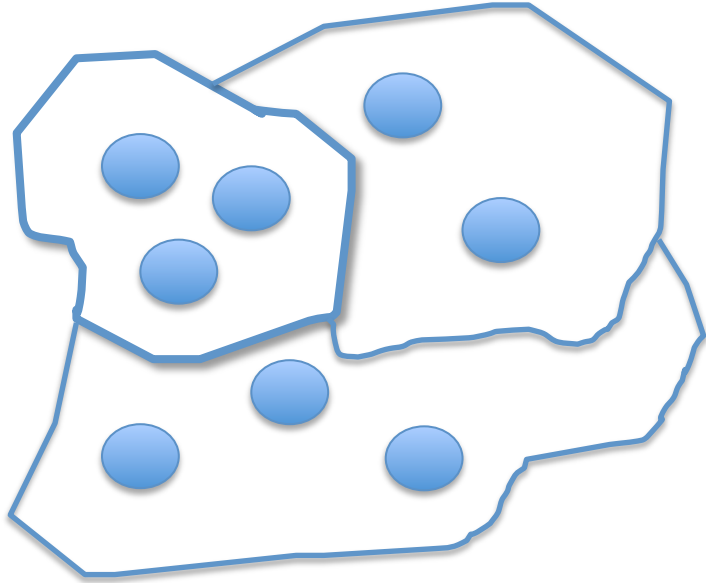
# Dealing With Complexity: Microservices

Monoliths are the enemy -- Circa 2012

It's ok to have copies! Use the tool that fits

Database per service

Pub/Sub Event Queues



Multiple services per bounded context



MachineHealth.com



# Dealing with Complexity

## Event Sourcing the new old frontier

What events need to move across your company's context boundaries?

# Thank you!

<https://oatscenter.org>

