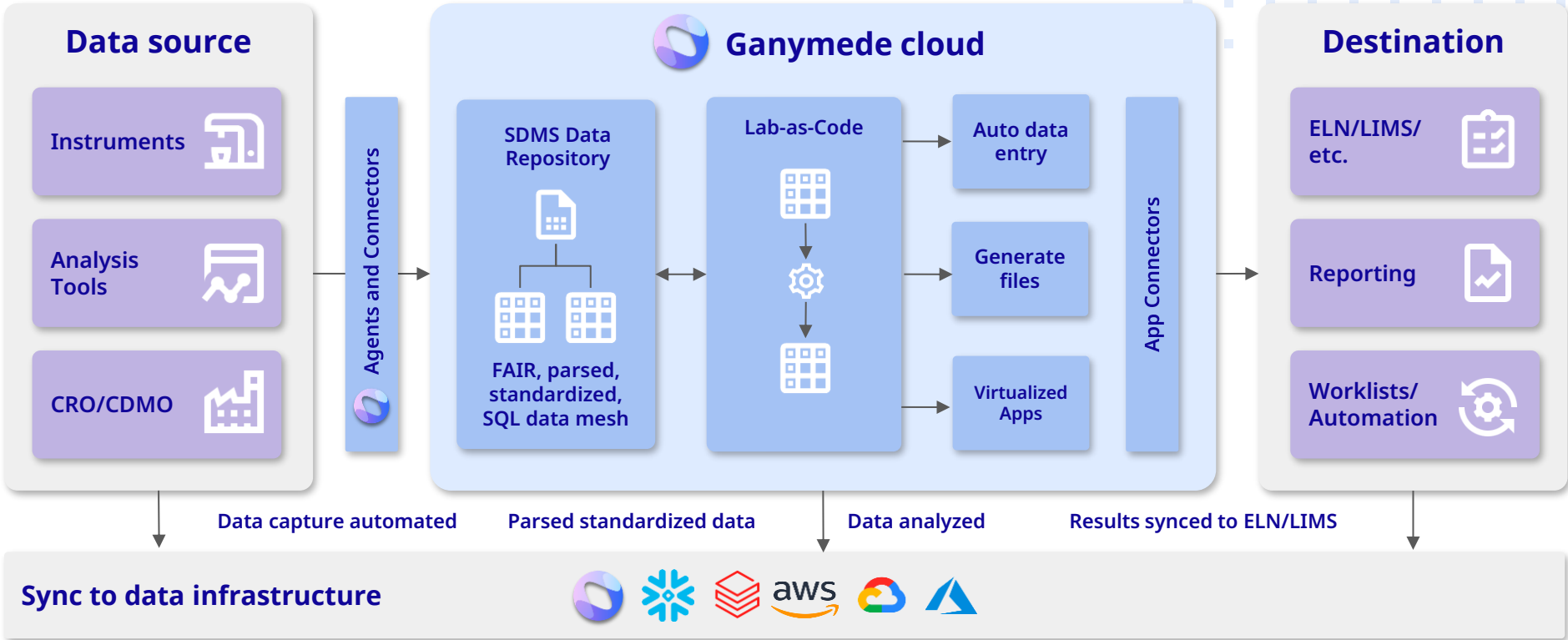




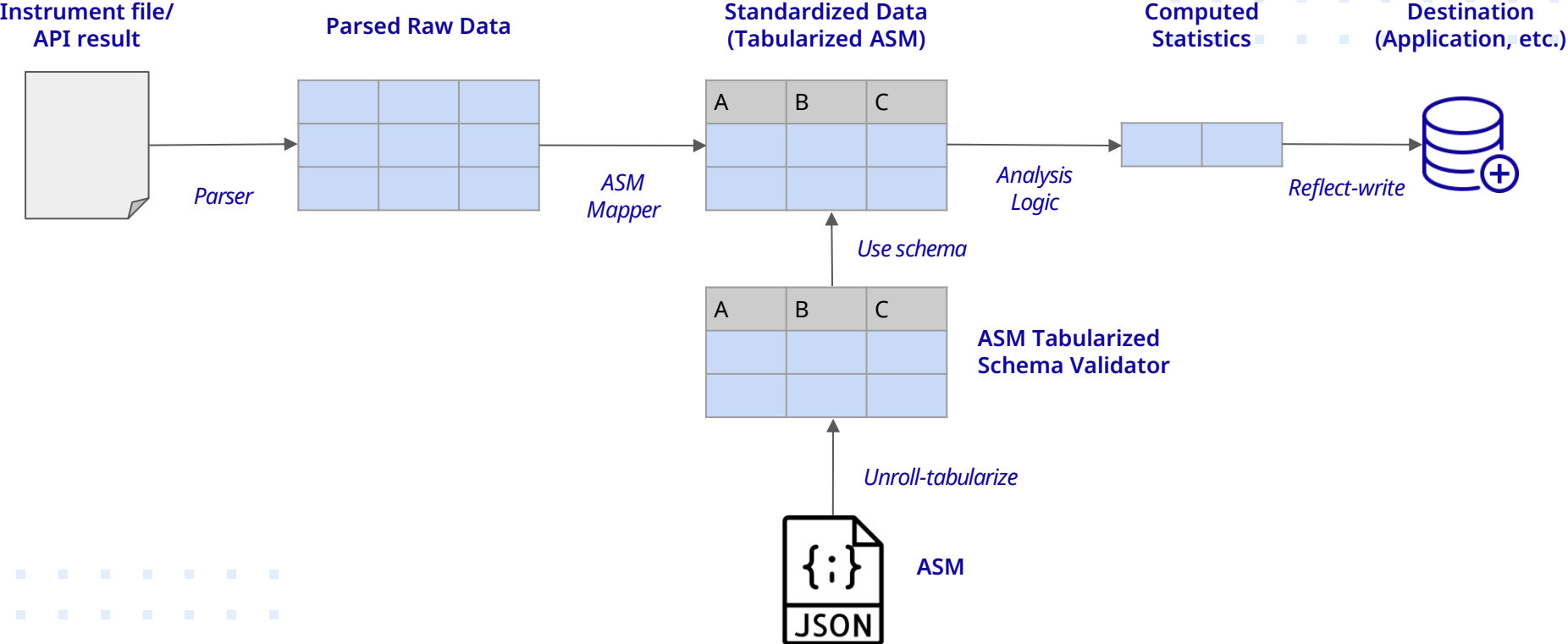
Ganymede

Ganymede platform and strategy

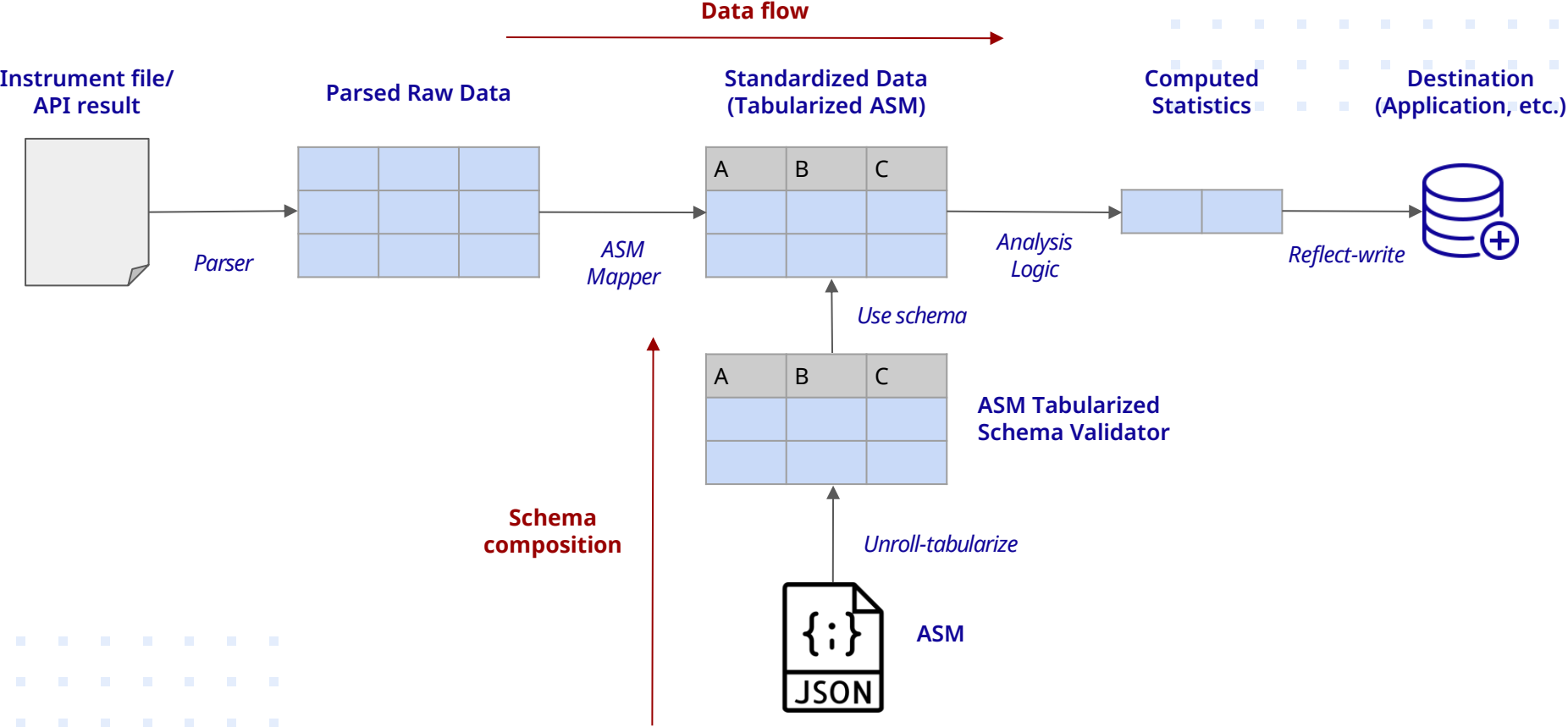
Lab-as-Code drives Ganymede's modular life sciences data integration and automation tools



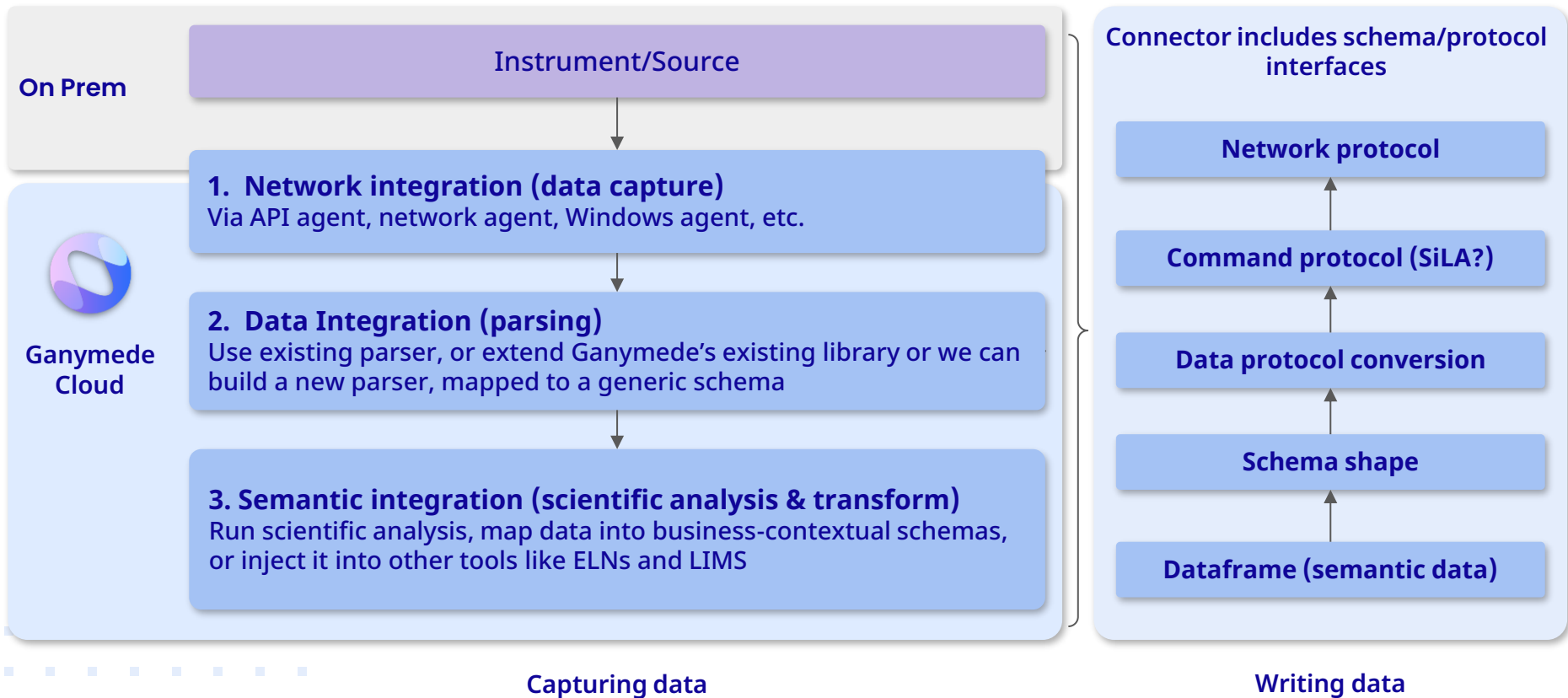
Ganymede adoption of Allotrope



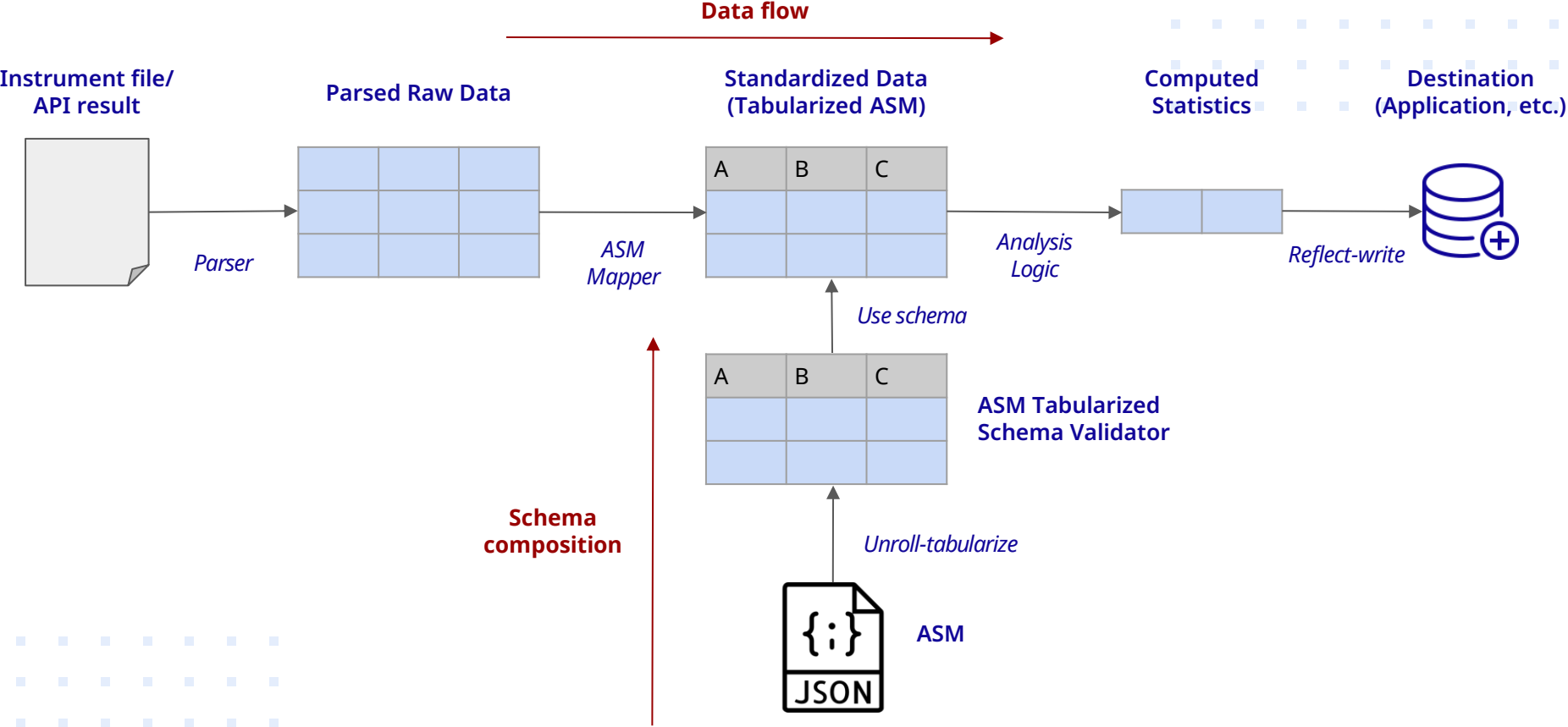
Ganymede adoption of Allotrope



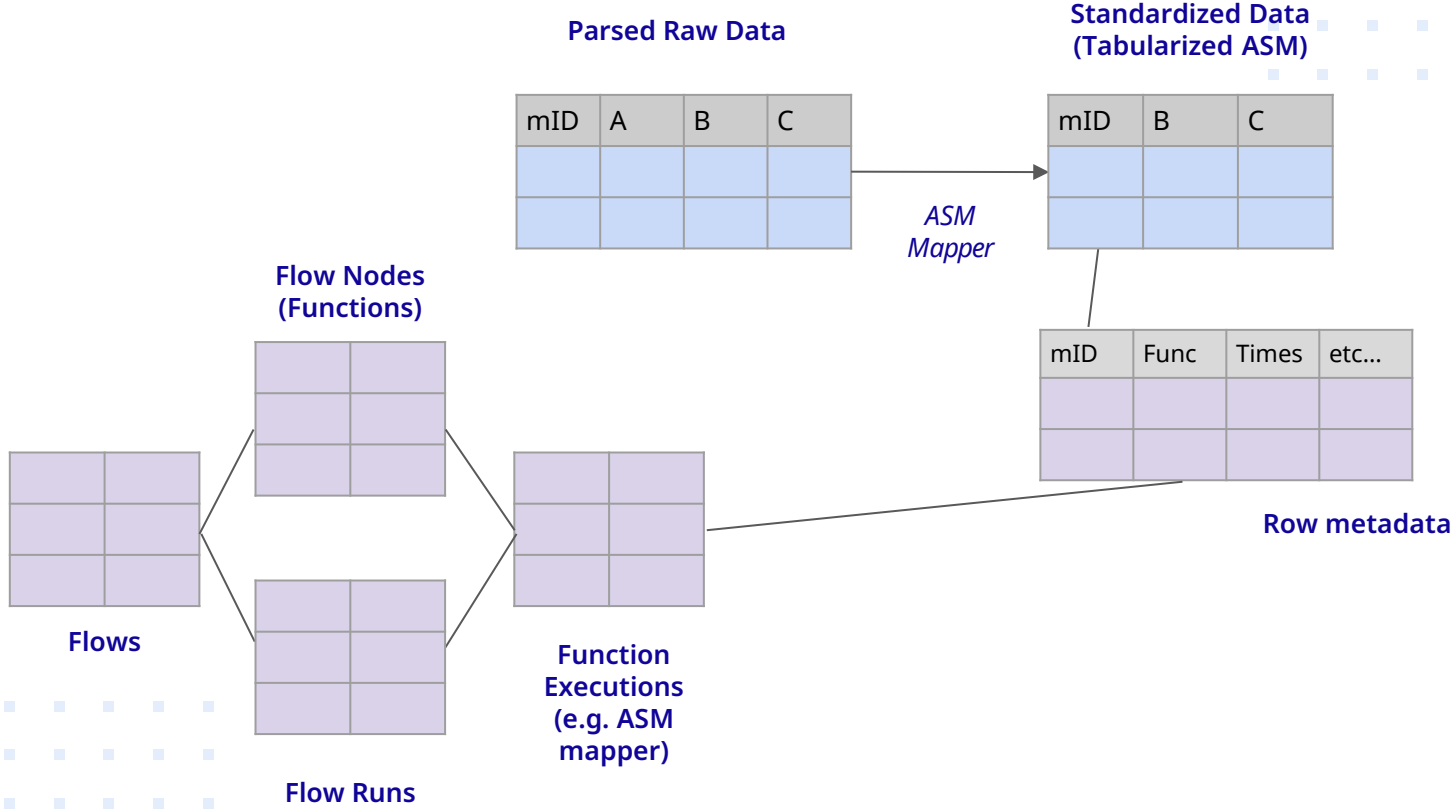
Interfaces in our integration paradigm



Data flow can be join conditions

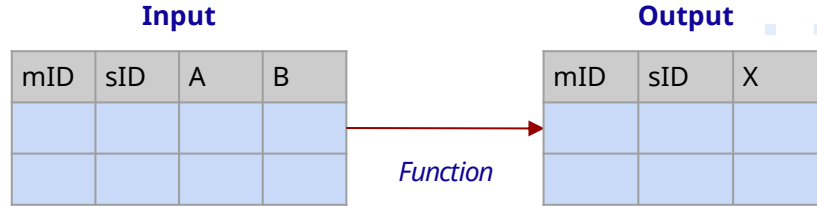


Example: Joining context across the ASM mapper

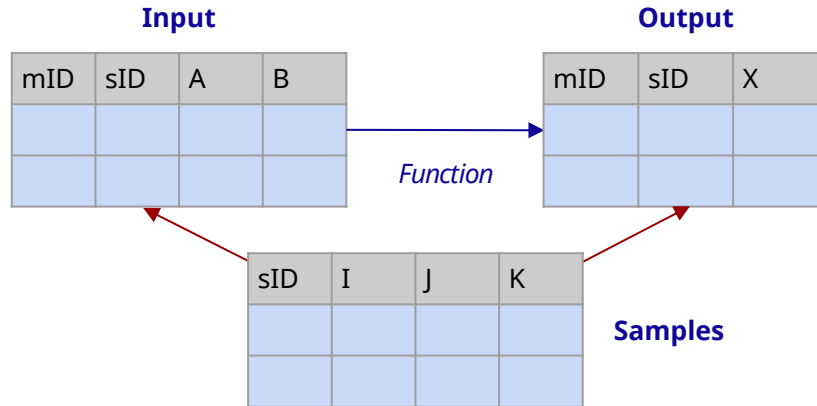


Threading context through functions

Join via functional relations (mID)

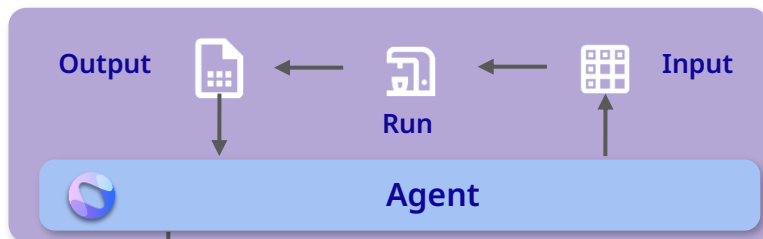


Join via sample associations (sID)



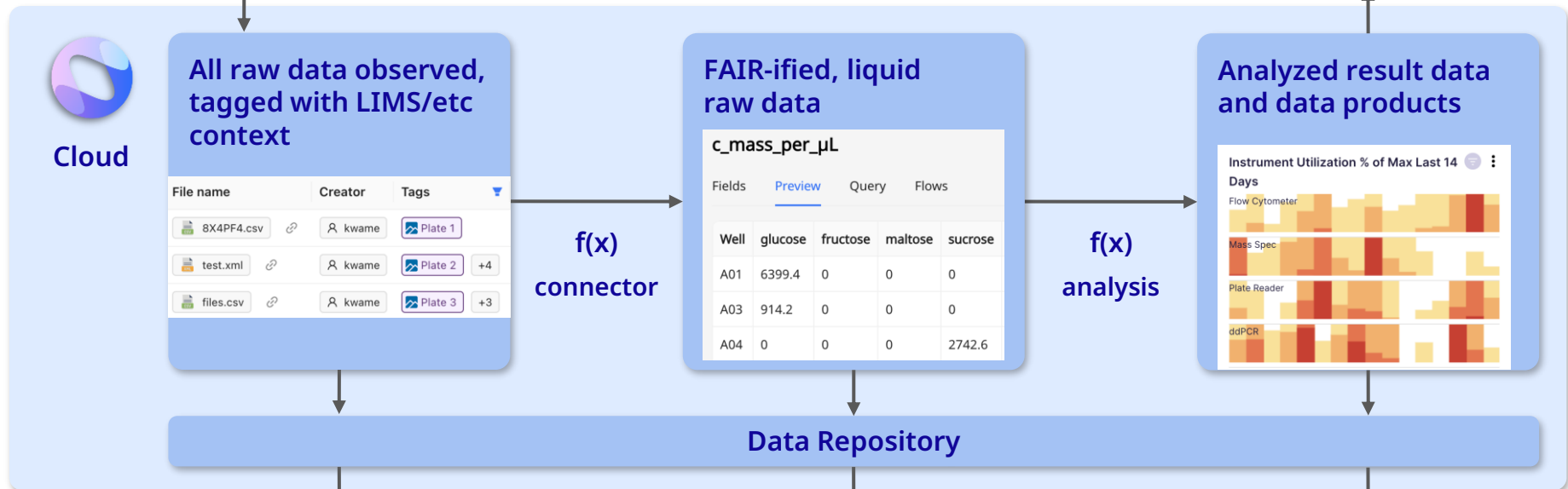
Full circular data integration

Instrument



trigger

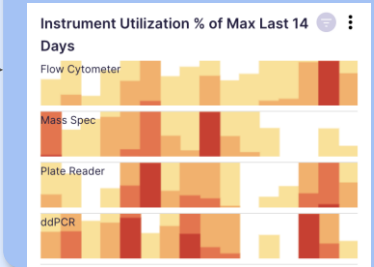
LIMS/MES/ELN/
Etc.



File name	Creator	Tags
8X4PF4.csv	kwame	Plate 1
test.xml	kwame	Plate 2 +4
files.csv	kwame	Plate 3 +3

c_mass_per_μL

Fields	Preview	Query	Flows	
Well	glucose	fructose	maltose	sucrose
A01	6399.4	0	0	0
A03	914.2	0	0	0
A04	0	0	0	2742.6



Sync to data infrastructure



Summary: our lab data integration principles

- 1. Joinability is the gold standard of traceability;** this applies to standards mappings:
 - a. Functions are usually implemented in a lossy way for context in simple systems
 - b. Implement standards as mappers in a functional, composable way to allow for traceability of data (to join context/upstream data)
 - c. Implement standards as mappers in pipelines for traceability of logic (because the mappers' inputs and outputs will change over time)
 - d. Business schemas are the same - how easily can you join business schema data to standardized scientific data?
 - e. Tables (which can be interfaces over non-tabular data, like cubes) best enable joinability
- 2. Circular LIMS/analysis assay automation is the gold standard of data integration** and maximizes context
 - a. What interfaces *consume* standardized data? Does your analysis pipeline?
- 3. Semantic layers should be thick/colocated; protocol layers should be thin/modular.** By layer:
 - a. Semantic data standards: maximize analysis automation (e.g. Allotrope) and sit side by side with your business/operational schemas in LIMS - good to make these "thick" layers
 - b. Data protocol standards: simpler is better to make this layer thin (e.g. flat files/JSON, dataframes.) Don't conflate databases with data; the distinction is growing (e.g. Apache Iceberg)
 - c. Action data standards: are ideally purely CRUD on data interfaces, therefore thin
 - d. Network protocol standards: are already fully abstracted (REST, files, events, etc.)

Impact: a lower barrier to Allotrope adoption

- Allotrope is “pulled in” naturally by interfaces that consume it like analyses
- Allotrope can be used selectively and composed with other structures (like business schemas)
- More things can be mapped into Allotrope more easily
- Mapper development can be more agile because mappings are managed better