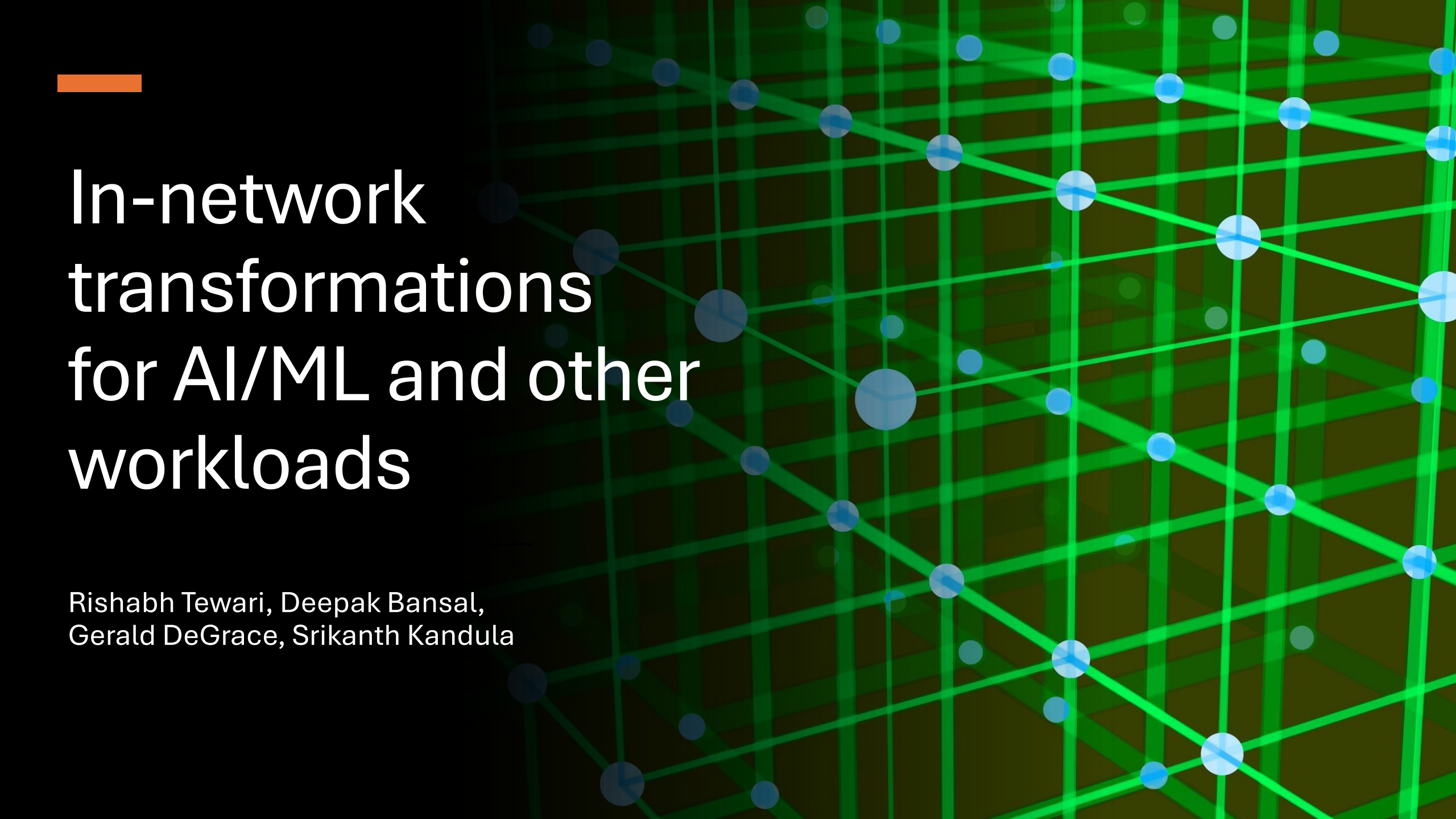


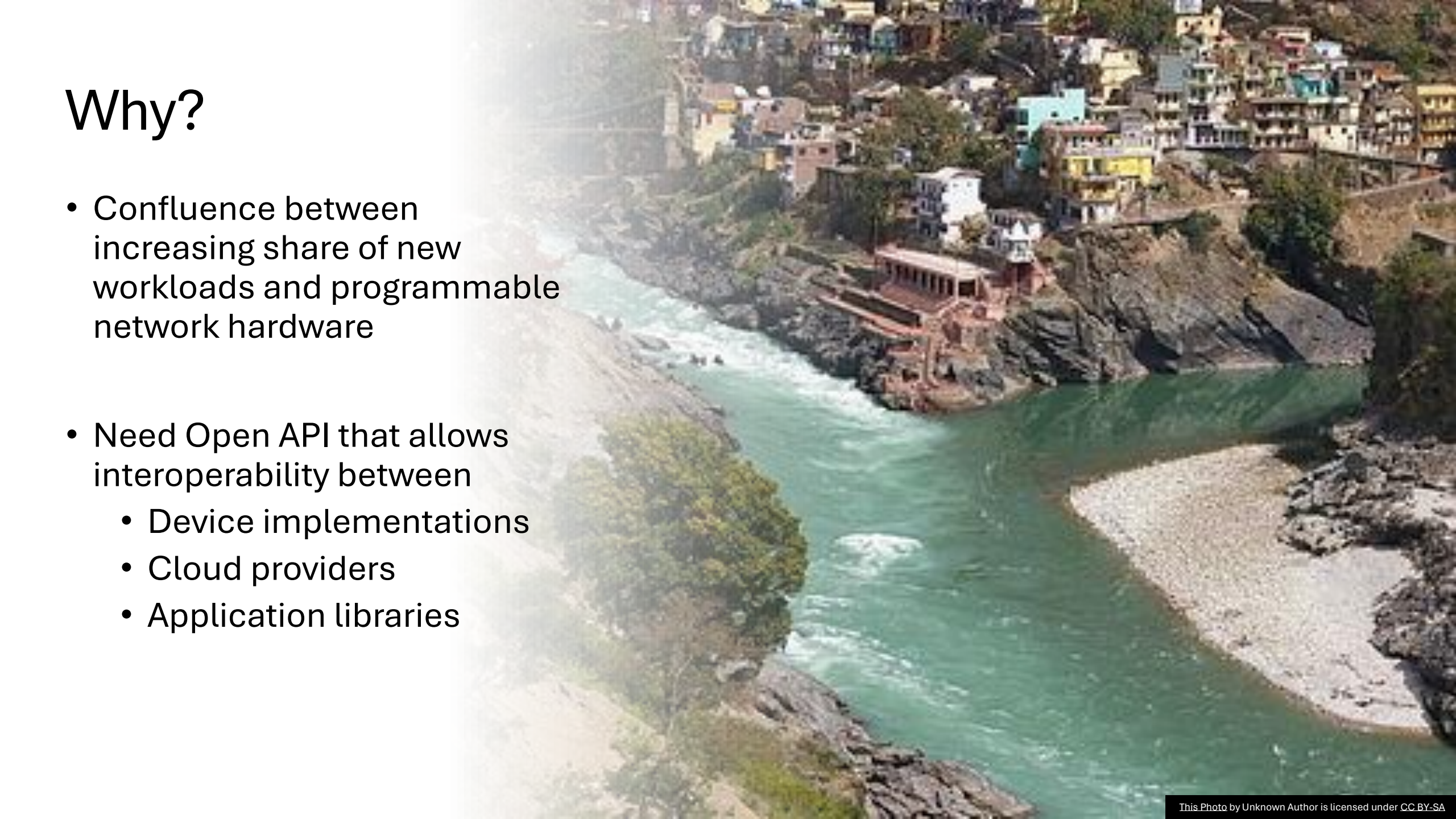
In-network transformations for AI/ML and other workloads

Rishabh Tewari, Deepak Bansal,
Gerald DeGrace, Srikanth Kandula

Why?

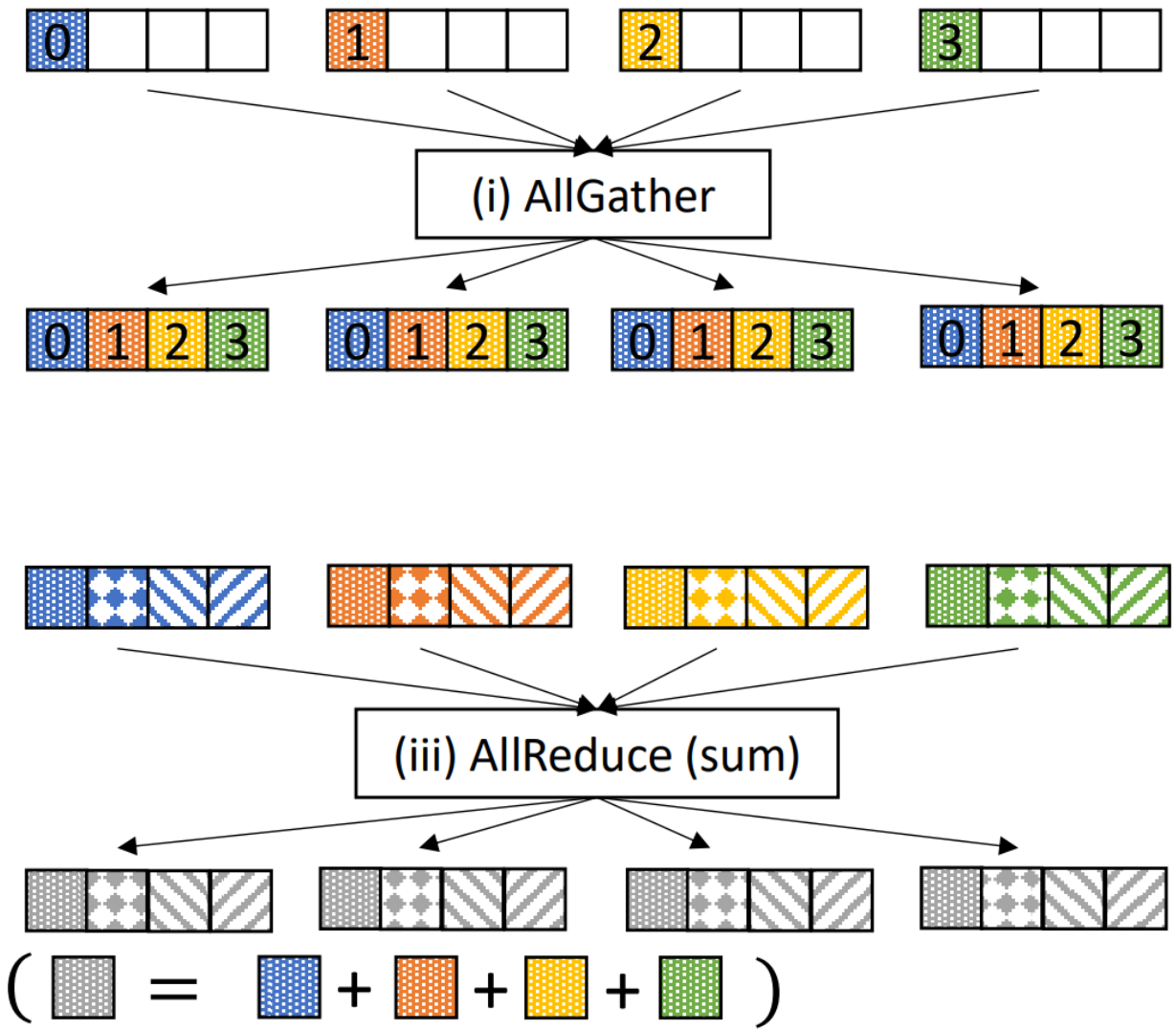
- Confluence between increasing share of new workloads and programmable network hardware

- Need Open API that allows interoperability between
  - Device implementations
  - Cloud providers
  - Application libraries



Which transformations?

- Multicast
- In-network aggregations
- Gradient compression
  - Sparse transforms
  - Floating point changes
- Future proof



Transform Abstractions

- One-to-one (e.g., gradient compression)
- One-to-many (e.g., multicast)
- Many-to-one (e.g., in-network aggregation)

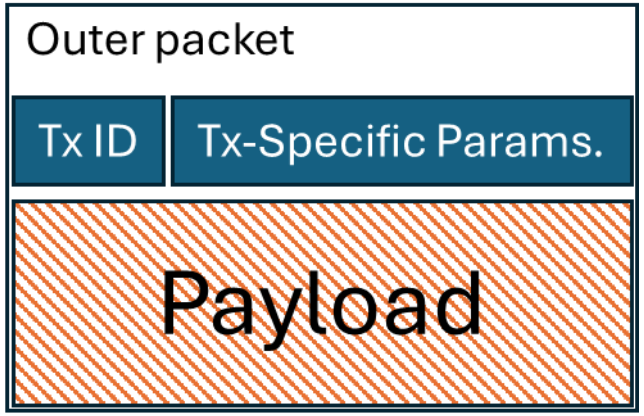
Only many-to-one is stateful (requires payload state)

Can chain multiple transforms
* e.g., many-to-many

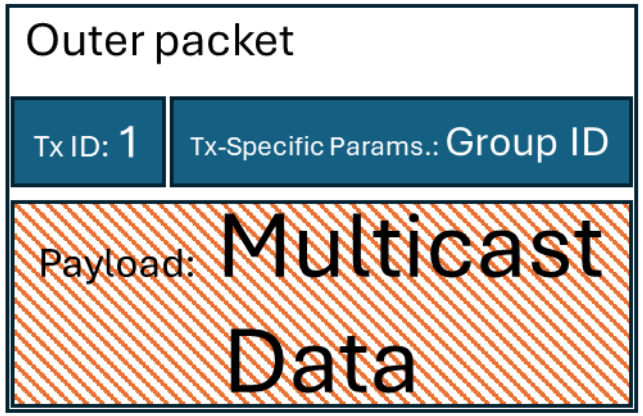
Design questions

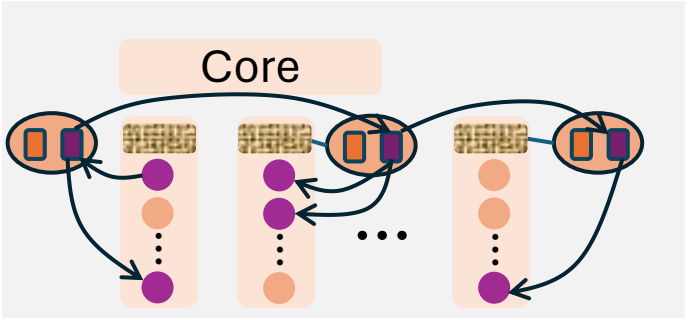
- Carefully orchestrated paths, aggregation points vs. Disaggregated and opportunistic
  - sHarp vs. ATP
- Coexist with multiple L2, L3, L4 protocols vs. specialization

An initial stab at protocol

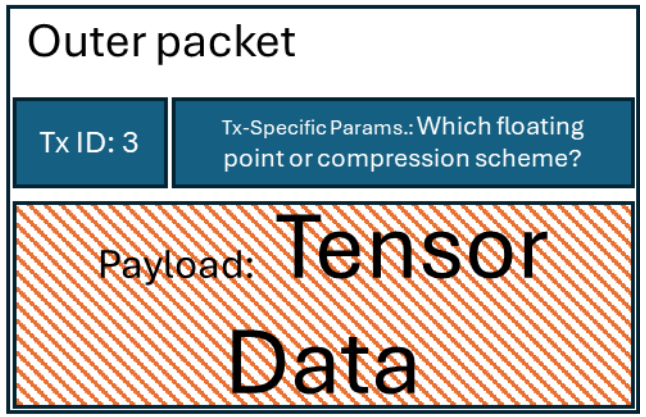


One-to-many Example

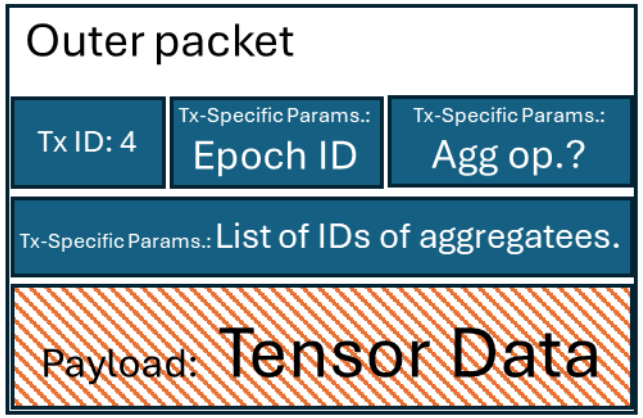


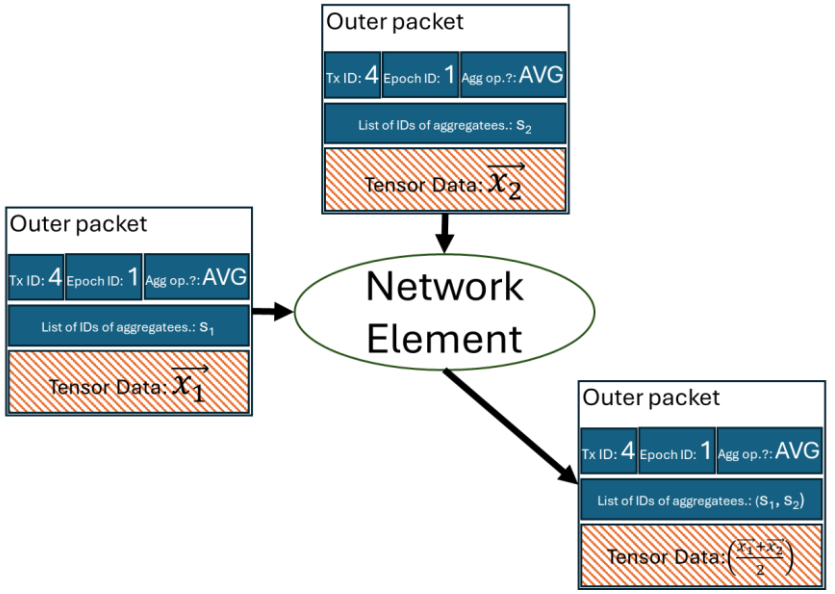


One-to-one example



Many-to-one


Outer packet
Tx ID: 4
Tx-Specific Params.:
Epoch ID
Tx-Specific Params.:
Agg op.?
Tx-Specific Params.: List of IDs of aggregatees.
Payload: Tensor Data


Outer packet
Tx ID: 4
Epoch ID: 1
Agg op.?: AVG
List of IDs of aggregatees.: S_2
Tensor Data: $\overrightarrow{x_2}$
Outer packet
Tx ID: 4
Epoch ID: 1
Agg op.?: AVG
List of IDs of aggregatees.: S_1
Tensor Data: $\overrightarrow{x_1}$
Network Element
Outer packet
Tx ID: 4
Epoch ID: 1
Agg op.?: AVG
List of IDs of aggregatees.: (S_1, S_2)
Tensor Data: $\left(\frac{\overrightarrow{x_1}+\overrightarrow{x_2}}{2}\right)$