

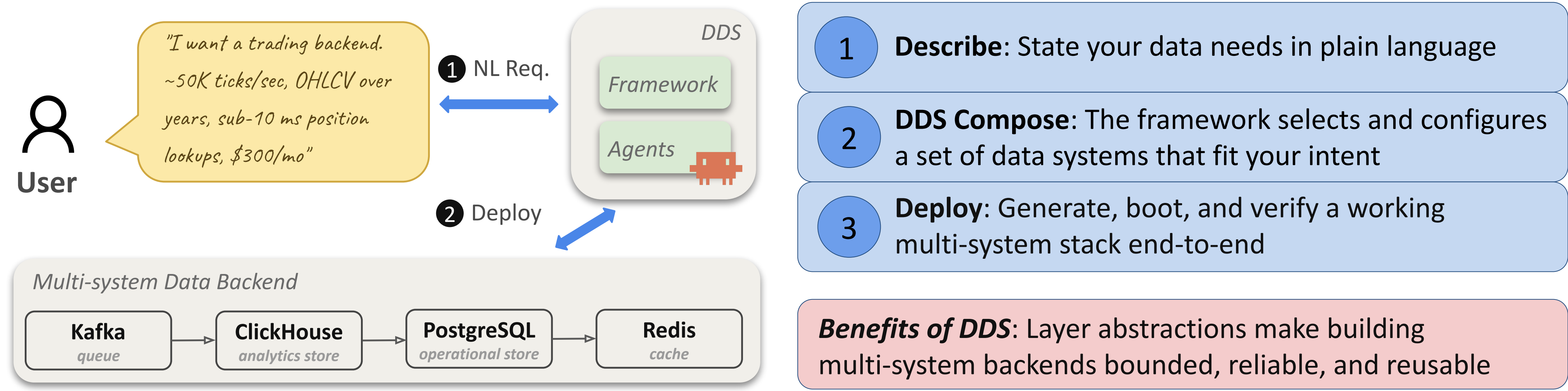
Declarative Data Services: Structured Agentic Discovery for Composing Data Systems

Shanshan Ye, Duo Lu

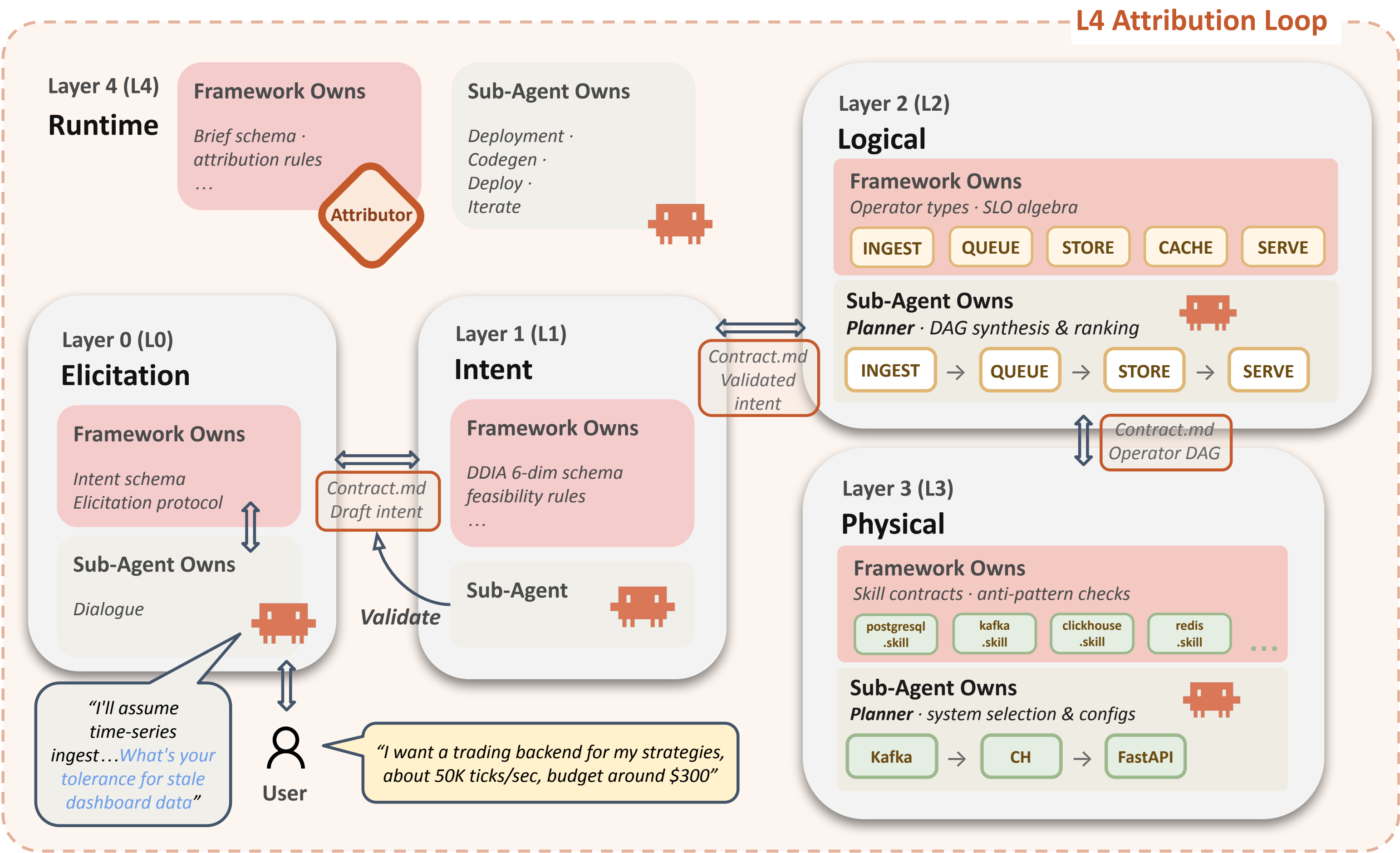


What? Give users a runnable multi-system data backend from a natural-language prompt
Why? Unbounded coding agents fail to converge on heterogeneous multi-system stacks
How? Typed contracts bound the search so failures route to the layer that owns them

End-to-End Workflow of Declarative Data Services (DDS)



DDS Framework: Four Typed Contract Layers



- Key Design # 1:**
The framework owns every contract; sub-agents own the search inside it
- Key Design # 2:**
Skills accumulate fixes across deployments so a patch made once carries forward
- Key Design # 3:**
L4 routes each failure to the layer owning the violated decision

Experimental Evaluation: DDS vs. Unbounded Agentic Discovery

Condition	n	T0	T1	T2	Median \$
A: NL only + 5-iter loop	10	10/10	2/10	2/10	\$5.76
B: NL + system names + 5-iter loop	10	10/10	3/10	2/10	\$6.23
C: + skill YAMLs (prose) + 5-iter loop	10	10/10	6/10	4/10	\$5.06
DDS: full framework, one-shot L4	10	9/10	8/10	3/10	\$1.49
DDS: full framework + 5-iter loop	10	10/10	10/10	10/10	\$1.94

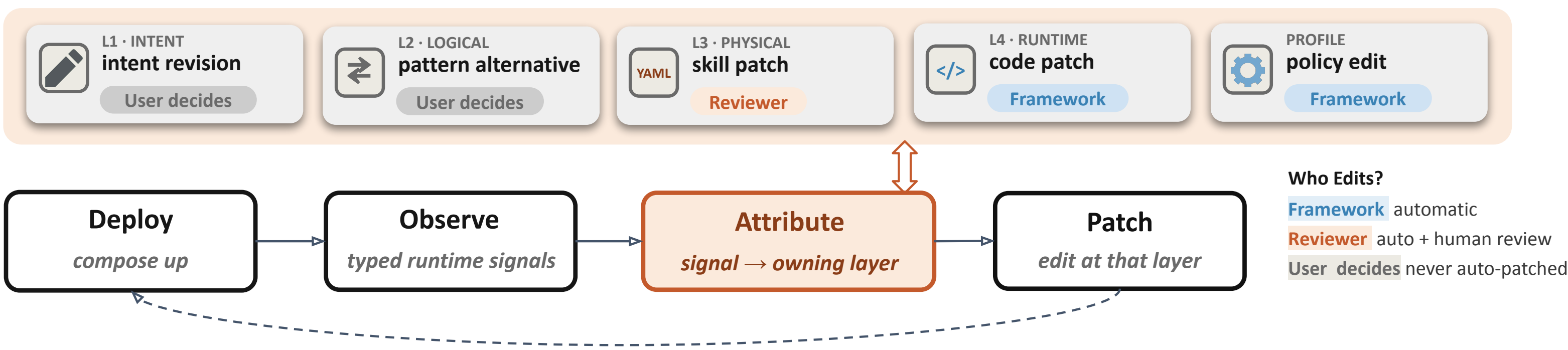
Table 1: Headline Comparison

Skill variant	n	T0	T1	T2	Median \$
full (all four blocks)	3	3/3	2/3	2/3	\$1.36
ops-stripped (drop operational + anti_patterns)	3	3/3	0/3	0/3	\$1.11
minimal (capabilities only)	3	3/3	0/3	0/3	\$0.65

Table 2 : Skill-Content Ablation

- Three tiers: T0 (syntax), T1 (stack boots), T2 (a test query returns correct rows)
- Conditions A/B/C: unbounded agents with progressively more prompt knowledge
- DDS: typed contracts at all four layers

Table 1 Highlights:
Condition C has the same skill knowledge as DDS but fails to converge on T1/T2 without typed contracts
Table 2 Highlights:
operational + anti_patterns blocks do the heavy lifting



Typed attribution is why iteration converges. Each failure routes to exactly one layer
The fix is bounded: a skill patch, a code fix, or a user decision, never a global retry