CEF 2026 · VENICE · PART 2

Lessons from other fields

How other computational communities became interoperable — and what economics should copy

Akshay Shanker (Econ-ARK) · **Matt McKay** (QuantEcon)

Second talk of the session: the diagnosis (Part 1), the precedent (Part 2), the program (Part 3).

Why this part exists

ONE CHAIN OF EVIDENCE

Part 1 says the economics code base is fragmented. Part 2 asks whether other fields solved the same problem, and which piece of their solution is portable.

1

Packages are not enough

Reusable code helps only when papers agree on the object the code preserves.

2

Model and method are conflated

The same economics often arrives welded to a solver, grid, interpolation choice, or simulation convention.

3

No shared meaning layer

We lack a common statement of the dynamic model before numerical realization begins.

4

Incentives matter

Institutions explain adoption; the PI's opening carries that part. Here we isolate the technical object.

Start with the economics fact

THE PROBLEM IS SILOING, NOT COMPETENCE

CURRENT ECONOMICS CENSUS	VALUE
Code-found papers	298
Solution-method diversity	5.47
Field-tool reuse rate	22%
Reinvention indicator	80%

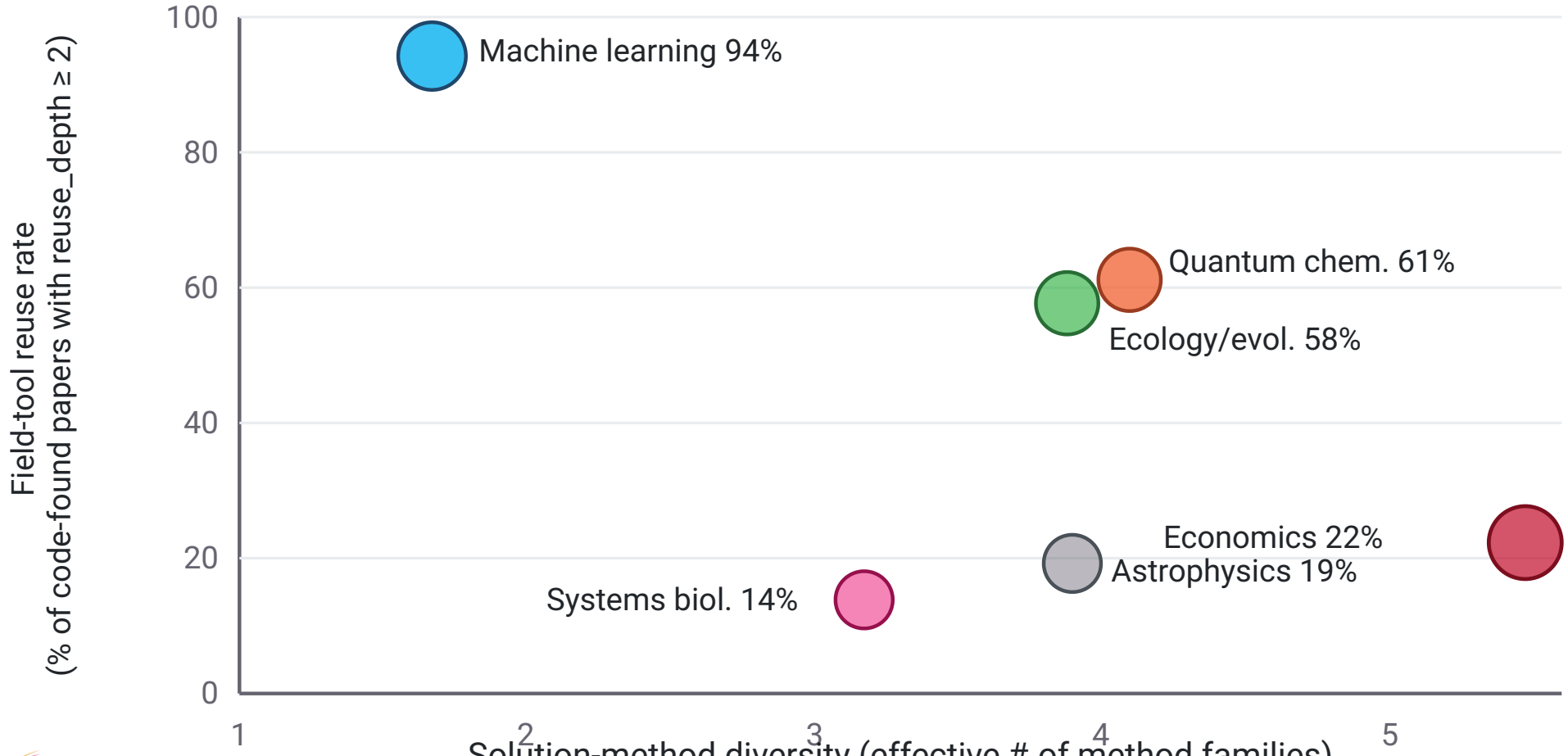
> The striking evidence is life-cycle siloing: closely related recursive problems are solved in separate code islands, with little borrowing across HARK, VFI Toolkit, GDSGE, SSJ, QuantEcon, or bespoke code.

Dynare is the exception economists already recognize: when the class is clear and the tooling is valuable, economists do coordinate.

Internal preliminary census: crossfield_prelim.csv and all_field_cards.md. Field-tool reuse rate is the share of code-found papers with reuse_depth ≥ 2.

Diversity and field-tool reuse come apart

MEASURED FIELD DATA, CORRECTED METRIC



Positions are measured from the current preliminary field census. Colour annotates the kind of common layer; it is not interoperability depth as a coordinate.

The same map says where the common layer sits

LEVEL OF SHARED ABSTRACTION

FIELD	DIVERSITY	FIELD-TOOL REUSE	WHAT IS COMMON?	INTERPRETATION
Machine learning	1.67	94%	frameworks, architectures, foundation models	common layer is high and operational, not model-meaning
Quantum chemistry	4.10	61%	private/public engines plus I/O conventions	reuse by reference implementation and workflow discipline
Ecology/evolution	3.88	58%	R package and data-analysis layers	useful package reuse, shallow model agreement
Astrophysics	3.90	19%	deep engines, often private	low shared reuse is consistent with strong private engines
Systems biology	3.17	14%	SBML model artifacts and COPASI ecosystem	model-file infrastructure exists but does not dominate observed papers
Economics	5.47	22%	NumPy, BLAS, general numerics; Dynare island	common layer is low, beneath diverse methods and applications

> The lesson is not only how much a field shares. It is where sharing enters the stack.

What does the standard hold fixed?

A DIAGNOSTIC BEFORE BORROWING ANALOGIES

CASE	WHAT VARIES	SHARED OBJECT	MEANING INDEPENDENT OF ONE SOLVER?	WHAT ECONOMICS LEARNS
Dynare	calibration, shocks, DSGE variants	`.mod` statement and solution routines	partly, inside a clear class	start with the in-discipline precedent
Stan	priors, likelihoods, data, inference settings	probabilistic program / target log density	yes, bounded by differentiability and HMC-shaped tooling	model statement before inference
MiniZinc / AMPL / GAMS	instances and solver backends	math-program statement	yes, for known optimization classes	state the problem before naming a solver
SBML	mechanisms, parameters, simulation tools	model artifact for reaction networks	partly, with curated tests and validators	model exchange can be real infrastructure

All four narrow the scientific object. None is a generic package manager.

Dynare: Adjemian et al. 2011. Stan: Carpenter et al. 2017. MiniZinc: Nethercote et al. 2007. AMPL: Fourer et al. 1990. SBML: Hucka et al. 2003, 2019.

Reuse rate and interoperability depth are different objects

DO NOT PUT THEM ON THE SAME AXIS

INTEROPERABILITY DEPTH — A LADDER, NOT A NUMBER

SHALLOW — DATA
a shared file format or data convention

INTERFACE
coupling APIs and package contracts

REFERENCE IMPLEMENTATION
one dominant code or engine

MODEL ARTIFACT
a portable statement tools consume

MEANING LAYER
a compositional mathematical object

PROOF OBJECT
a kernel-checked logical term

Measured rate: share of code-found papers with field-tool reuse, using reuse_depth ≥ 2.

Interpretive depth: what kind of thing the community shares. It is an annotation: colour, side strip, or label.

> Lean shares at the logical-kernel level. ML shares at the framework and architecture level. Economics mostly shares below the model, at the numerical-substrate level.

Level language follows the interoperability-layers reading of Wang, Tolk, and Wang 2009, adapted for computational modelling.

Dynare first: the counterexample inside economics

STANDARDIZATION WORKS WHEN THE CLASS IS CLEAR

What Dynare fixed

- a DSGE model file
- perturbation and simulation workflows
- a community of examples
- enough shared syntax that papers can be compared

What it did not try to fix

- arbitrary heterogeneous-agent recursion
- discrete-continuous choice
- changing state spaces and branching
- mixed local methods and custom simulation loops

> Dynare is not a failure of standardization. It is evidence that scope discipline is the condition for adoption.

Adjemian et al. 2011; Dynare manual.

Three useful splits, each deliberately bounded

PRECEDENTS, NOT TEMPLATES

Stan

S

Write a probabilistic model; inference can vary. The split is powerful because the target density is the shared object.

Optimization languages

O

State a math program; choose a solver later. This is the cleanest model-method split in the comparison set.

SBML

B

Exchange a reaction-network model artifact; validators and test suites make it operational.

Limit

L

Each field wins by narrowing the class. Economics cannot copy the surface syntax and ignore that narrowing.

Carpenter et al. 2017; Nethercote et al. 2007; Fourer et al. 1990; Hucka et al. 2003, 2019.

Reuse without a meaning layer: two warnings

DO NOT CONFUSE ENGINE REUSE WITH MODEL AGREEMENT

Quantum chemistry

The field reuses engines and workflow conventions at a high rate. QCSchema and QCArchive make inputs, results, and provenance portable.

Warning: this is not a portable meaning layer for new economic models. It is a disciplined engine-and-record ecosystem.

Astrophysics

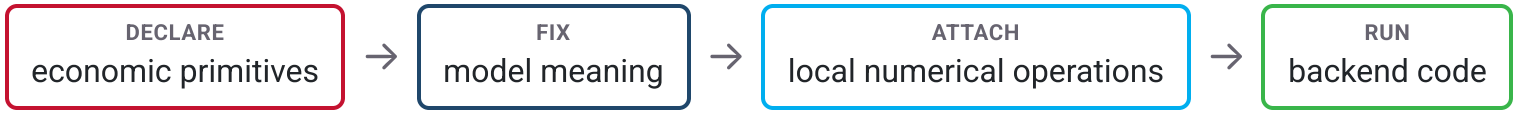
Many papers depend on deep engines such as simulation codes, but the engines are often private, customized, or not reused as shared field tools.

So the 19% shared-reuse rate is not a contradiction. It is what private-engine science looks like in this metric.

The object we need to separate

MODEL STATEMENT VERSUS NUMERICAL REALIZATION

CHANGE IN A PAPER	EXAMPLE	SHOULD A SHARED LAYER TREAT THIS AS THE SAME MODEL?
Parameters and settings	calibration, shock process, tolerances, grid density	usually yes
Model statement	preferences, timing, state variables, constraints, equilibrium closure	no, the economics changed
Numerical realization	interpolation, quadrature, endogenous grid, policy iteration, simulation backend	yes if it computes the same declared object



The reusable object is not the whole paper's model. It is the recurring operation structure that tells us whether two computations target the same dynamic problem.

The split is hard; that is exactly the point

METHOD WORK CAN BE ECONOMIC SUBSTANCE

Present-biased agents. Moving from exponential to quasi-hyperbolic discounting is not a new option flag; it changes the recursive object and can force new solution logic.

Higher-dimensional endogenous grids. A richer state space is not merely a larger grid; it can change which operator is numerically usable.

Equilibrium closure. A household block reused in partial equilibrium may not have the same role once prices, aggregates, and market clearing are endogenous.

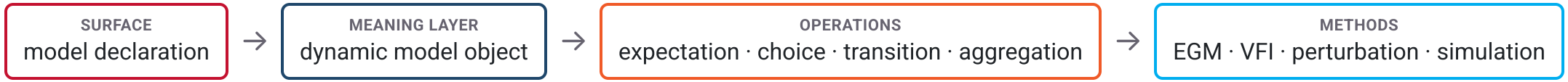
Therefore: the meaning layer must separate what it can, and expose where the separation fails.

> The standard is not a promise that methods become automatic. It is a way to say, precisely, what the method is attached to.

Carroll 2006 on endogenous-grid methods; Auclert et al. 2021 on sequence-space Jacobians; internal Bellman-DDSL Rosetta Stone spec.

What a meaning layer means for economics

RAISE THE COMMON LAYER, BUT NOT TO FULL MODELS



The SQL analogy

SQL separated the logical query from the physical plan. The economics analogue is not "tables"; it is separating the declared dynamic object from the numerical plan.

The stack claim

DDSL should raise economics' common layer from arrays and solvers to the model-meaning and operation level: above the numerical substrate, below the complete model.

Codd 1970; Guagliardo and Libkin 2017; Bellman-DDSL Rosetta Stone spec.

The self-critique has to be sharp

THE STANDARD CAN FAIL IN ORDINARY WAYS

RISK	WHY IT IS SERIOUS	DISCIPLINE REQUIRED
Dynare and HARK are real precedents	economists already have successful shared tools	explain the missing object, not novelty for novelty's sake
Too deep becomes unusable	a rich language can price out the papers it should help	standardize the thinnest layer that still fixes model meaning
Infrastructure burden	principles without parsers, tests, reference models, and backends do not matter	build validation, examples, and migration paths from day one
Numerical accountability remains	equal model meaning does not imply equal accuracy	require diagnostics, tolerances, and explicit method contracts
Adoption is earned	weak mandates mean no one will use a standard on principle	make today's work easier before asking for community discipline

> A meaning layer without parser, tests, reference models, repository, and useful backends is irrelevant.

The revised claim

WHAT SURVIVES THE FIELD COMPARISON

> Standardize the smallest recurring operation grammar whose model meanings can be composed and audited. Use coupling interfaces so methods and backends can attach locally.

We will never have a scikit-learn for economics at the level of complete models — and we should not want one. We are not in the business of running the same model again and again. Reuse belongs at the method and operation level, not the full-model level.

NOT THIS	BUT THIS
one universal economics language	a shared meaning layer for recurring dynamic-model operations
one solver	multiple methods attached to named operations
reuse of complete models	reuse of household blocks, transitions, expectations, aggregators, and tests

Part 3: make the layer useful

HANDOFF

Econ-ARK: Bellman-DDSL

Declare the dynamic problem, fix model meaning, then expose named operations that HARK, JAX, Dynare-adjacent tooling, or custom solvers can consume.

QuantEcon: canonical repository

Maintain reusable benchmark models: Aiyagari, Krusell-Smith in two implementations, Buffer Stock, a simple life-cycle model, and simple SSJ/HANK examples.

> Part 3 is not a philosophical appendix. It is the engineering test: can these declarations become coupling interfaces that economists actually reuse?

References

Economics and internal evidence. Current preliminary field census: crossfield_prelim.csv; all_field_cards.md. Bellman-DDSL Rosetta Stone spec. Carroll 2006, "The method of endogenous gridpoints," https://doi.org/10.1016/j.econlet.2005.09.013. Auclert et al. 2021, "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models," https://doi.org/10.3982/ECTA17434. HARK docs, https://docs.econ-ark.org/.

Standards and modelling languages. Adjemian et al. 2011, "Dynare: Reference Manual, Version 4," https://www.dynare.org/wp-repo/dynarewp001.pdf. Carpenter et al. 2017, "Stan," https://doi.org/10.18637/jss.v076.i01. Nethercote et al. 2007, "MiniZinc," https://doi.org/10.1007/978-3-540-74970-7_38. Fourer, Gay, and Kernighan 1990, "A modeling language for mathematical programming," https://doi.org/10.1287/mnsc.36.5.519. Hucka et al. 2003, "The systems biology markup language," https://doi.org/10.1093/bioinformatics/btg015.

Cross-field infrastructure. Codd 1970, "A relational model of data," https://doi.org/10.1145/362384.362685. Guagliardo and Libkin 2017, SQL query meaning, https://doi.org/10.14778/3151113.3151116. MolSSI QCSchema, https://molssi.org/software/qcschema-2/. Smith et al. 2021, QCArchive, https://doi.org/10.1002/wcms.1491. FITS standard, https://fits.gsfc.nasa.gov/fits_standard.html. Astropy Collaboration 2013, https://doi.org/10.1051/0004-6361/201322068.

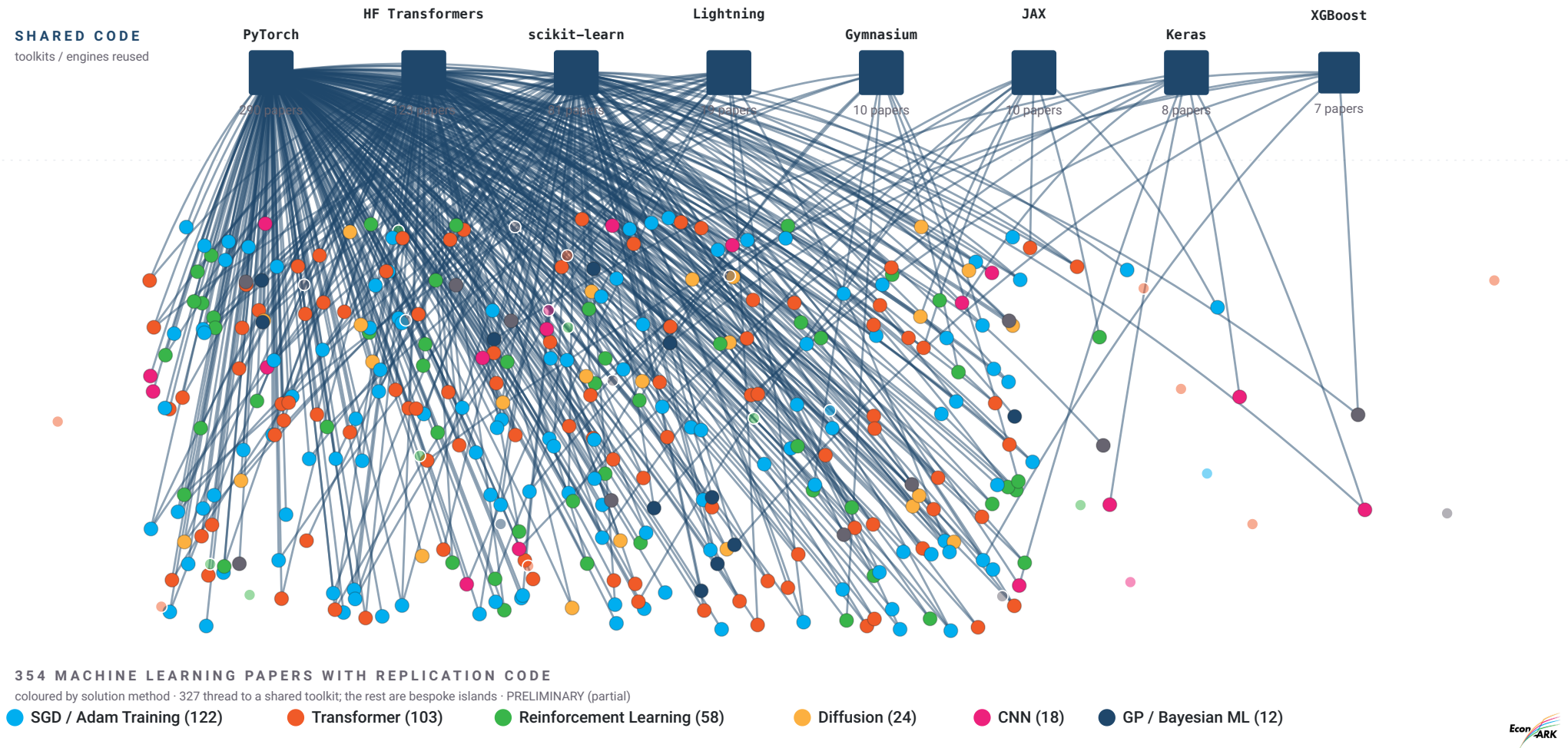
Levels and comparators. Wang, Tolk, and Wang 2009, "The levels of conceptual interoperability model," https://arxiv.org/abs/0908.0191. The mathlib Community 2020, "The Lean mathematical library," https://doi.org/10.1145/3372885.3373824. de Moura and Ullrich 2021, "The Lean 4 theorem prover and programming language," https://leanprover.github.io/papers/lean4.pdf. Buitinck et al. 2013, "API design for machine learning software," https://arxiv.org/abs/1309.0238.

APPENDIX · BACKUP

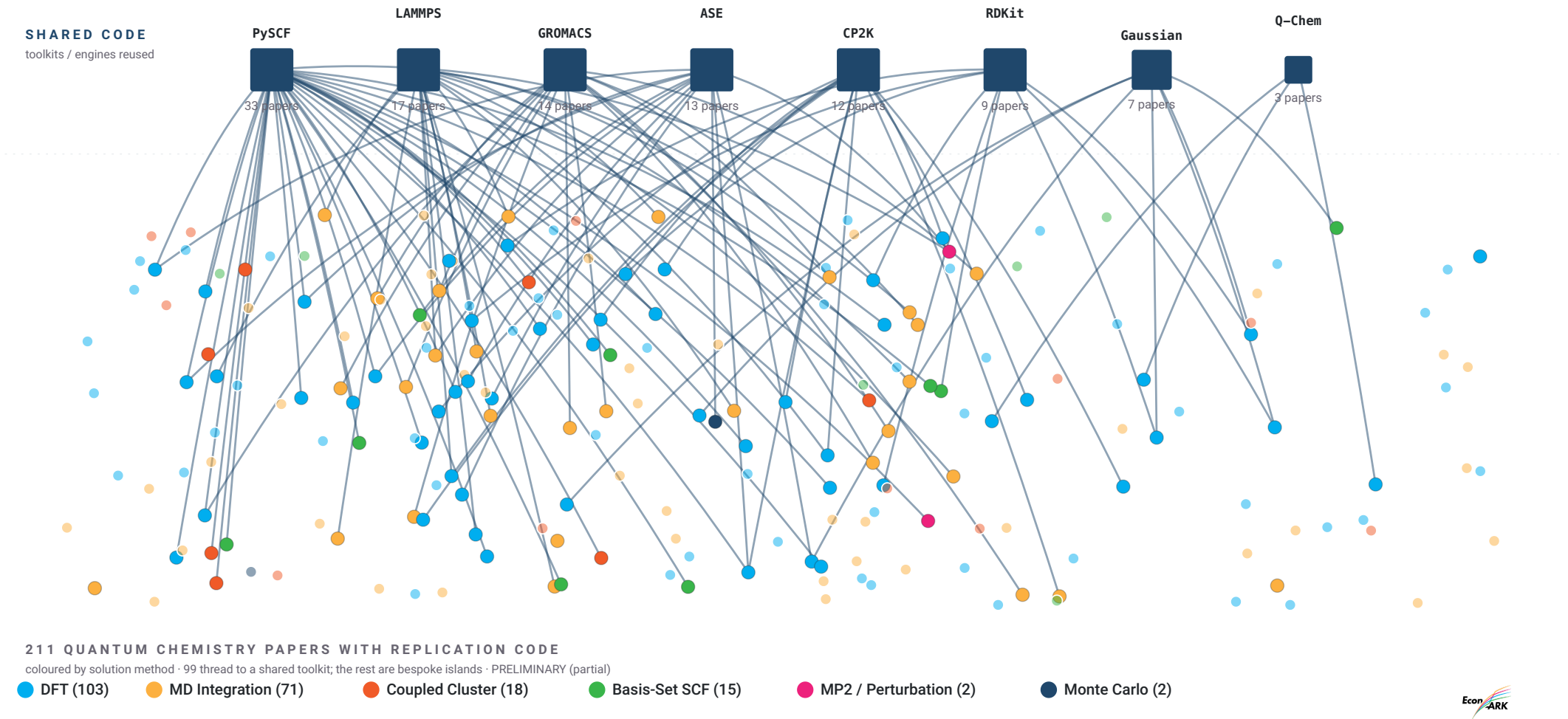
Reuse networks, field by field

Toolkits and engines actually reused, and the bespoke islands around them — preliminary census (~20% of papers).

Machine learning — shares the foundation, innovates above



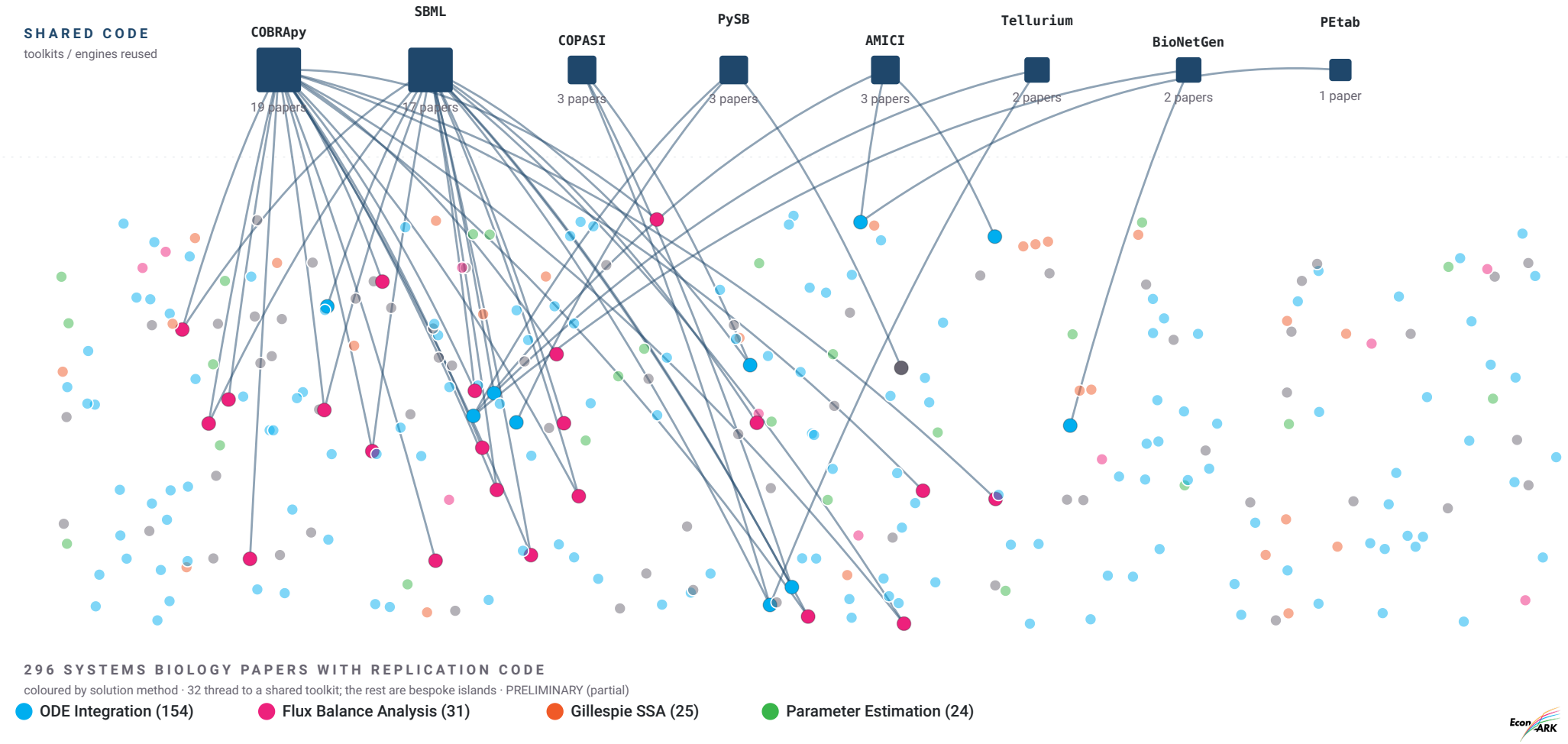
Quantum chemistry — reuses monolithic engines, innovates around them



Ecology & evolution — strong package reuse, some avoidable rebuild


Ecology and evolution reuse network

Systems biology — a substrate exists, but is underused



Astrophysics — deep engines, often private