Q*cert

A platform for specifying and verifying query compilers

J. Auerbach, M. Hirzel, L. Mandel, A. Shinnar, J. Siméon
IBM Research

Challenges?

Precise Language Semantics
Long Compilation Pipeline
Query Optimizer

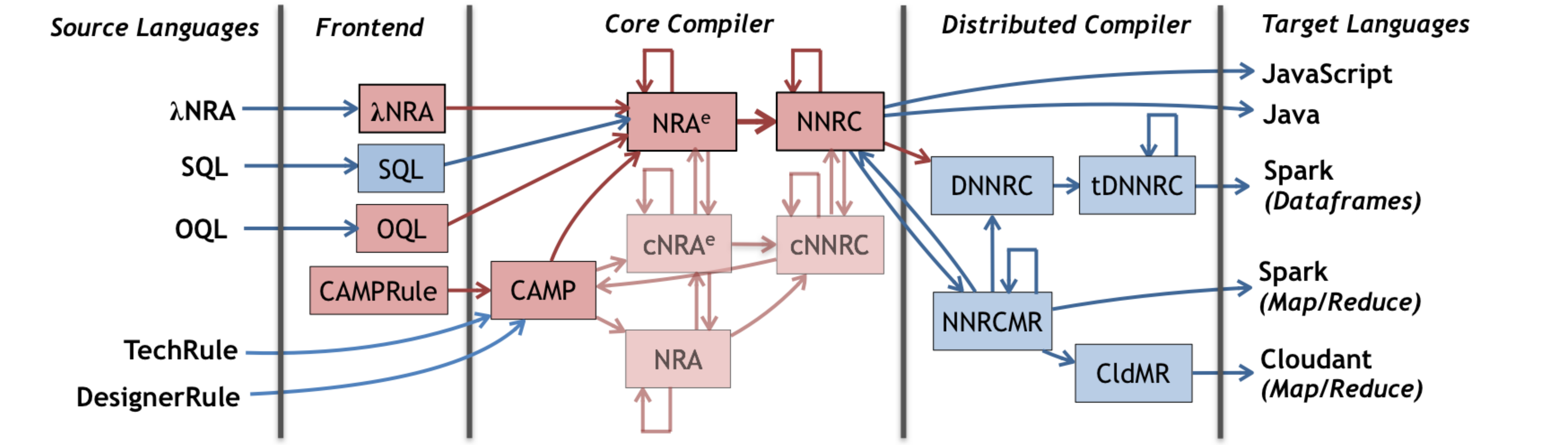
What for?

Correctness guarantees
New Languages (e.g., DSLs)
Education

How?

Formal Specification
Mechanized Proof
Code Extraction

Compilation Pipeline



Source Languages | Frontend | Core Compiler | Distributed Compiler | Target Languages

λNRA, SQL, OQL, TechRule, DesignerRule

λNRA, SQL, OQL, CAMPRule, CAMP

NRA^e, NNRC, cNRA^e, cNNRC, NRA

DNNRC, tDNNRC, NNRCMR, CldMR

JavaScript, Java, Spark (Dataframes), Spark (Map/Reduce), Cloudant (Map/Reduce)

SQL: Structured Query Language
OQL: Object Query Language
λNRA: NRA with Lambdas
CAMPRule: Rule Macros for CAMP
TechRule: ODM technical rules
DesignerRule: ODM designer rules

NRA: Nested Relational Algebra
NRA^e: NRA with Environments
cNRA^e: Core NRA^e
NNRC: Named Nested Relational Calculus
cNNRC: Core NNRC
CAMP: Calculus of Aggregating Matching Patterns

DNNRC: Distributed NNRC
tDNNRC: Typed DNNRC
NNRCMR: NNRC + Map/Reduce
CldMR: NNRC + Cloudant Map/Reduce

🟥 *with proof*
🟦 *without proof*

Algebraic Equivalence

```
Lemma select_union_distr q0 q1 q2 :
    σ⟨ q0 ⟩(q1 ∪ q2) ≡ σ⟨ q0 ⟩(q1) ∪ σ⟨ q0 ⟩(q2).
Proof.
  … (* proof omitted *)
Qed.
```

Functional Rewrite

```
Definition select_union_distr_fun q :=
  match q with
  | NRAEnvSelect q0 (NRAEnvBinop AUnion q1 q2) =>
      NRAEnvBinop AUnion
        (NRAEnvSelect q0 q1) (NRAEnvSelect q0 q2)
  | _ => q
  end.
```

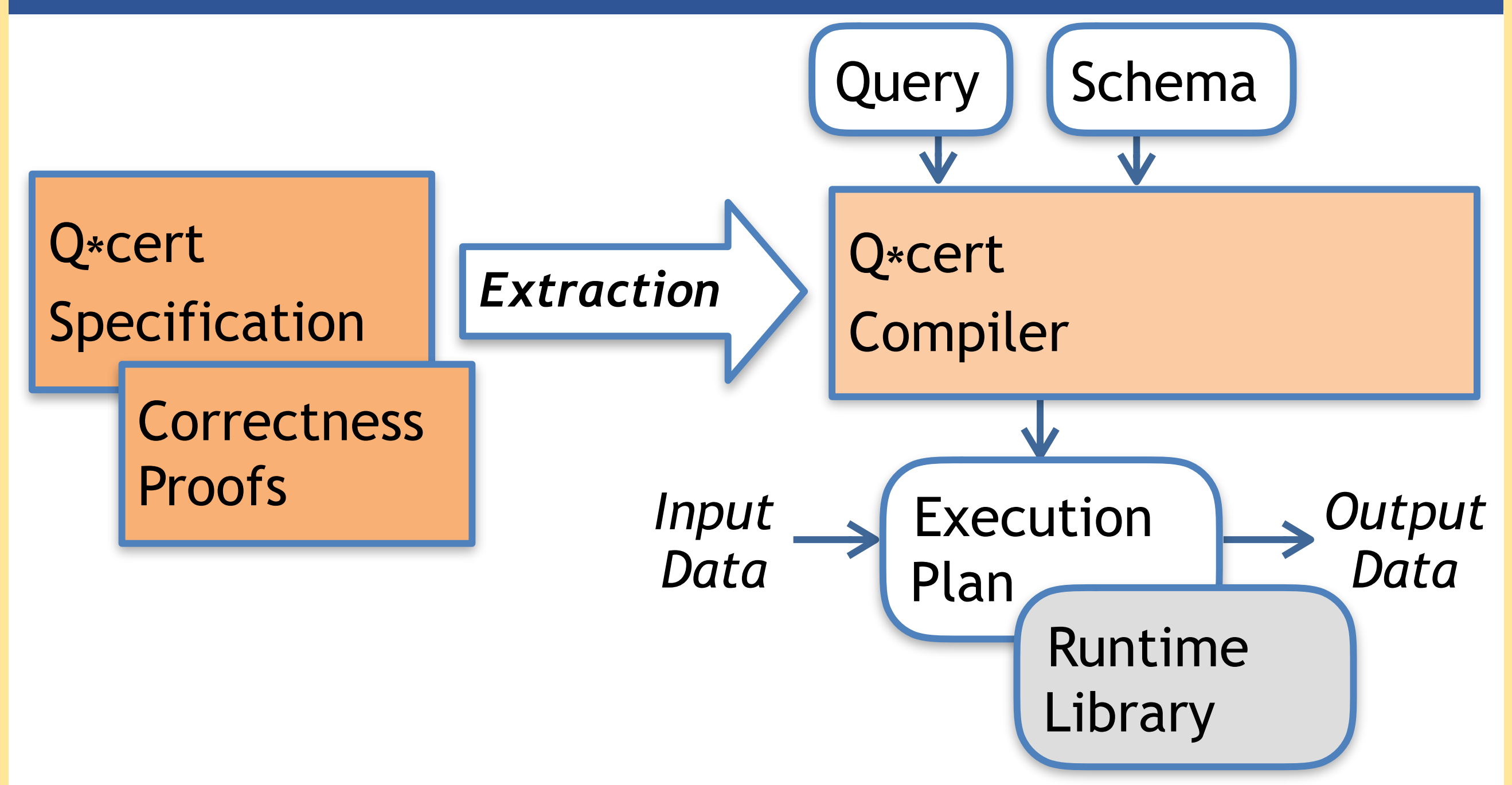
Correctness Proof

```
Property select_union_distr_fun_correctness q0 q1 q2 :
 select_union_distr_fun q ≡ q.
Proof.
 Hint Rewrite select_union_distr : envmap_eqs.
 prove_correctness q.
Qed.
```

Features

Nested Data Model with Objects
Type Checking
Aggregate Queries (includes TPC-H)
Configurable Optimizer
External Types and Functions
JSON Support

Compiler Extraction



Query, Schema

Q*cert Specification, Correctness Proofs

Extraction

Q*cert Compiler

Input Data, Execution Plan, Runtime Library, *Output Data*

The Coq Proof Assistant

https://querycert.github.io