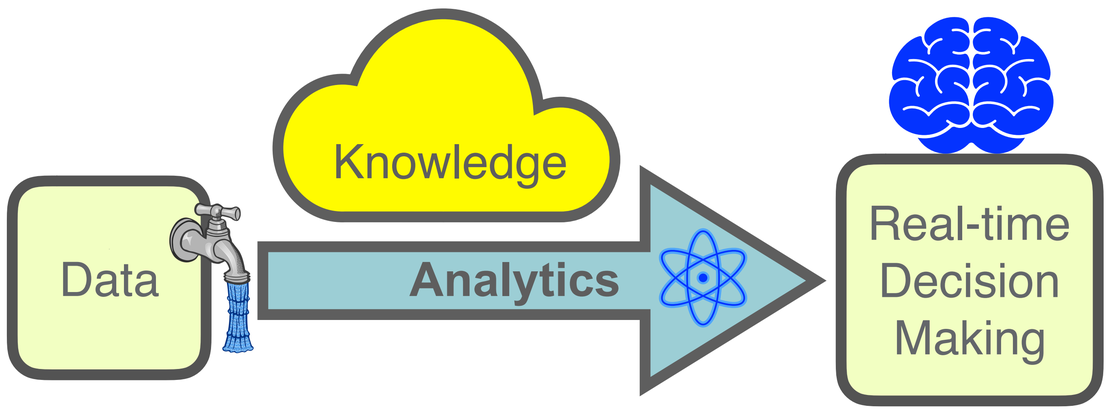
Continuous Deep Analytics (CDA)

Paris Carbone, Lars Kroll, Klas Segeljakt, Max Meldrum, Adam Hasselberg, Christian Schulte, Seif Haridi

<paris.carbone@ri.se> <lkroll@kth.se> <klasseg@kth.se> <mmeldrum@kth.se> <adamhas@kth.se> <cschulte@kth.se> <seif.haridi@ri.se>

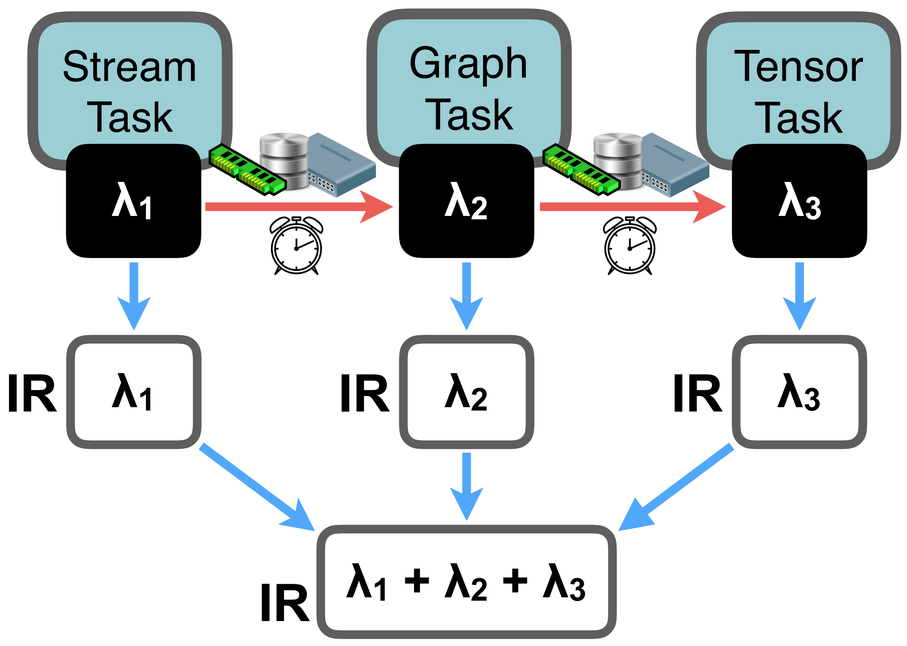
The Mission



The ultimate goal of the **CDA** project is to create a next-gen Big Data platform that can support **complex real-time decisions** based on **massive live data**.

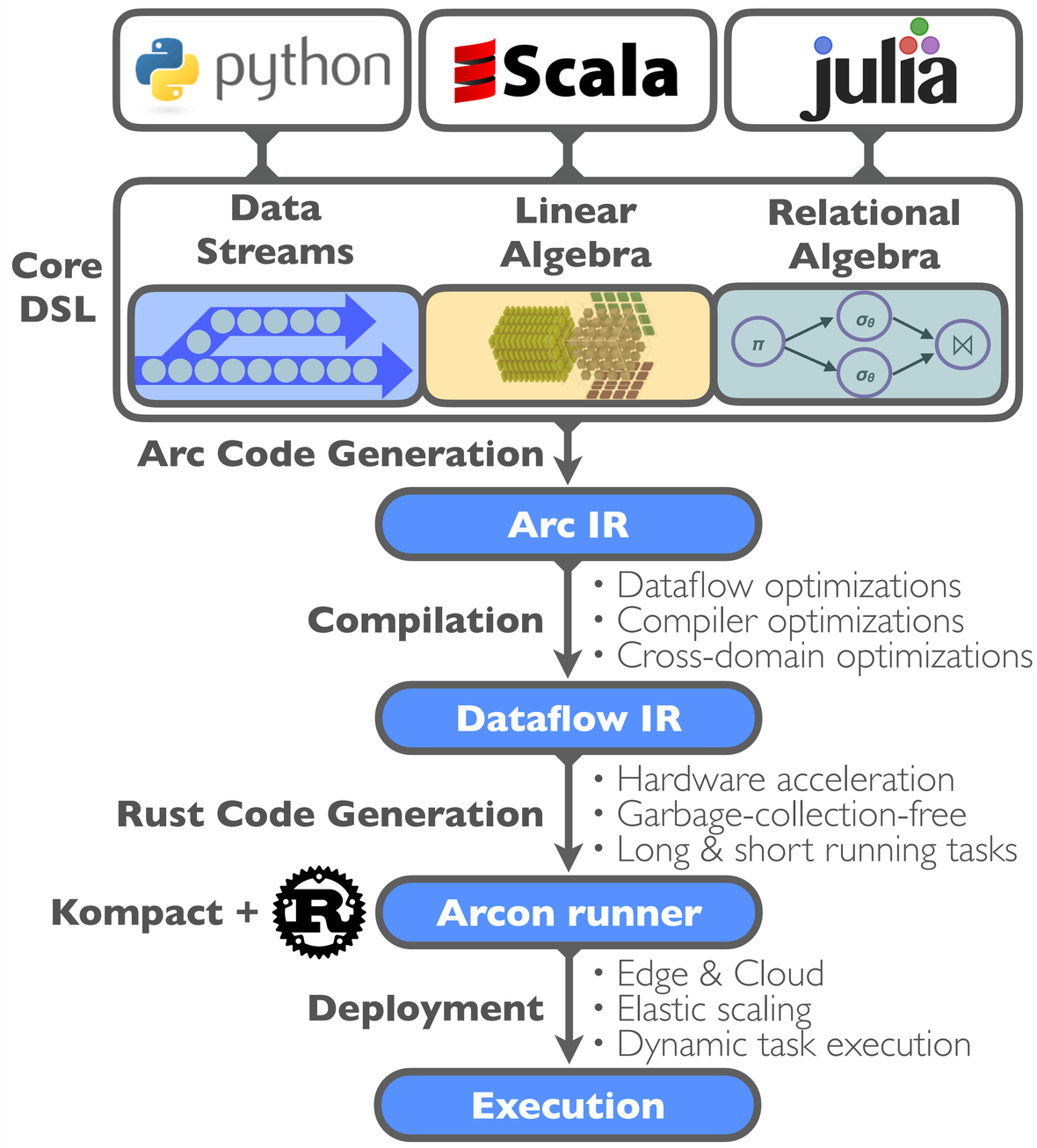
The Problem & Solution

Data analytics pipelines build on **diverse programming models** with hard abstraction boundaries. In effect, performance suffers from **context switching**, **steep data movement costs**, and **excessive type conversions**.



A solution is to **raise the level of abstraction** by introducing an **intermediate representation** (**IR**). The IR is a **programming language** that is able to **express** and **reason** about each of the programming models unitedly.

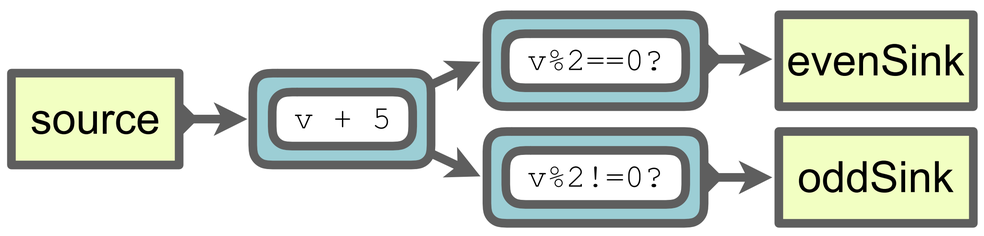
The CDA Stack



The CDA stack builds on four open-source projects:

- **Core DSL** - a **frontend** to the Arc IR, embedded in multiple host languages.
- **Arc** - a **programming language** for expressing and optimising computations that combine **data streams** with **relational** and **linear algebra**.
- **Arcon** - a **distributed runtime** which **Arc** runs **on**, implemented in **Rust**.
- **Kompact** - an event-based **component-actor** middleware used by **Arcon**.

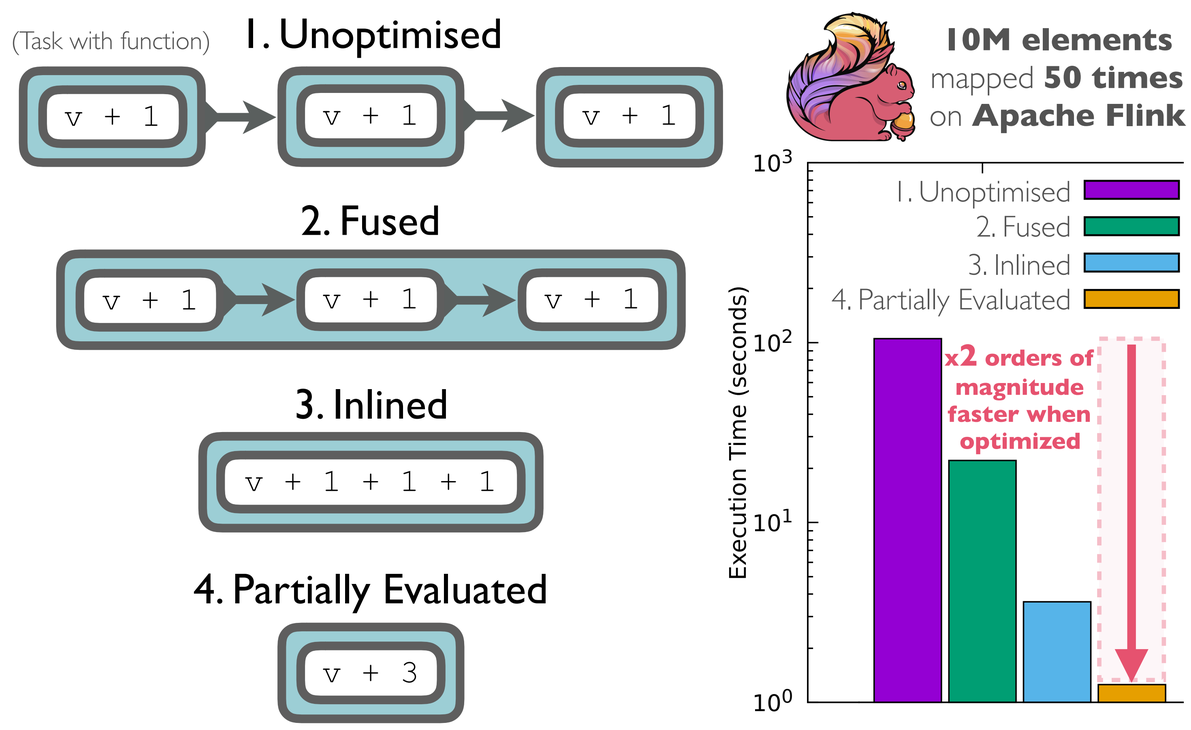
The Arc Intermediate Representation



Generated Arc code

```
|source:Stream[i32],
 evenSink:StreamAppender[i32],
 oddSink:StreamAppender[i32]|
 let mapped = result(for(source,
     StreamAppender[i32],
    |out, v| merge(out, v + 5)));
 for(mapped, evenSink, |out, v|
   if(v % 2 == 0, merge(out, v), out));
 for(mapped, oddSink, |out, v|
   if(v % 2 != 0, merge(out, v), out))
```

Performance



Looking for a MSc Thesis?

Scan this:

or contact klasseg@kth.se

Website: cda-group.github.io