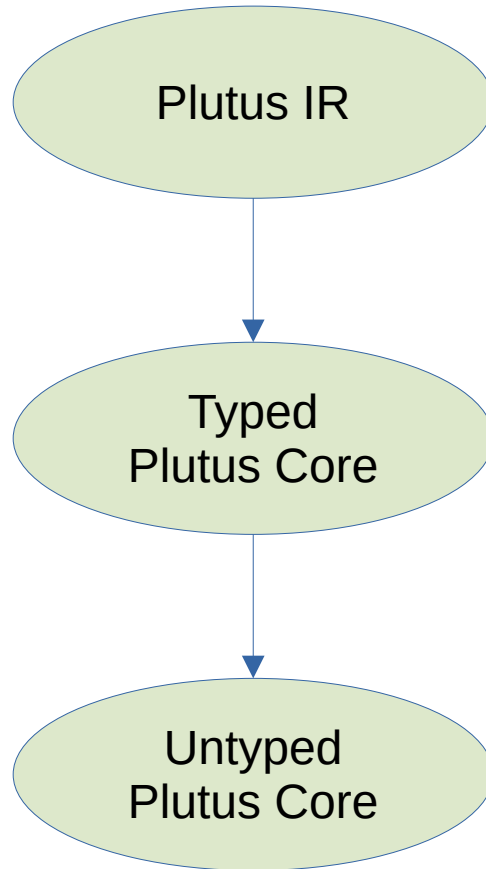


A unified *plutus* CLI tool

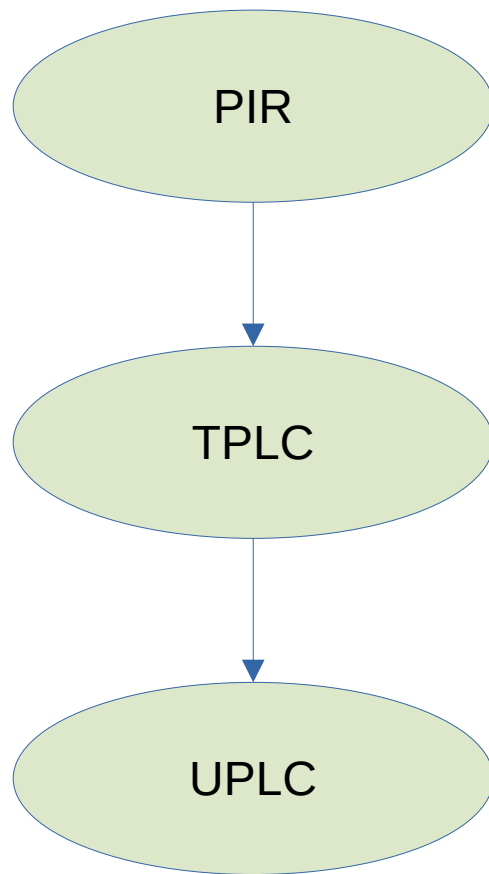
Nikolaos Bezirgiannis , Plutus Team

01 October 2024

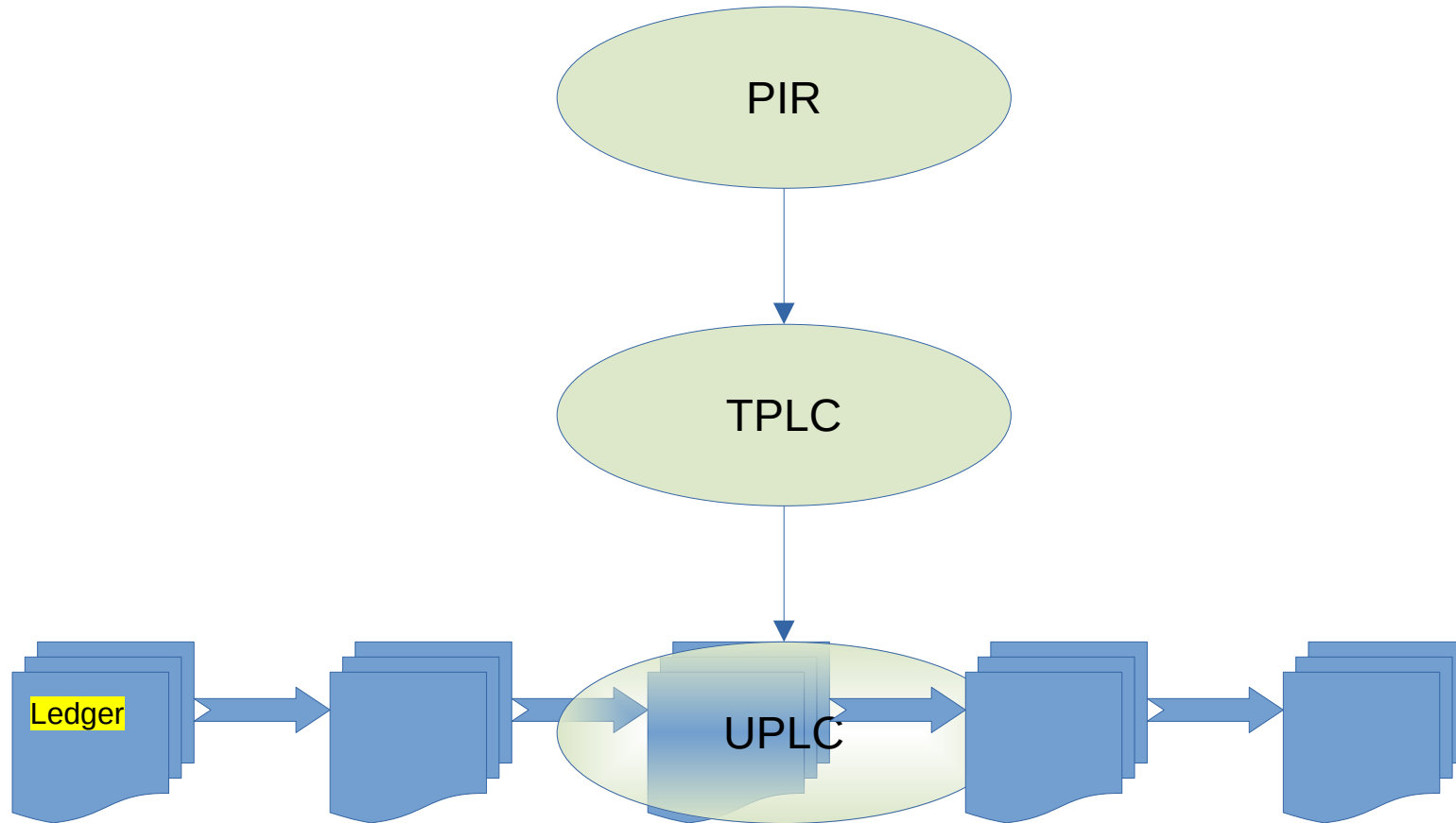
Intro: Compilation Pipeline



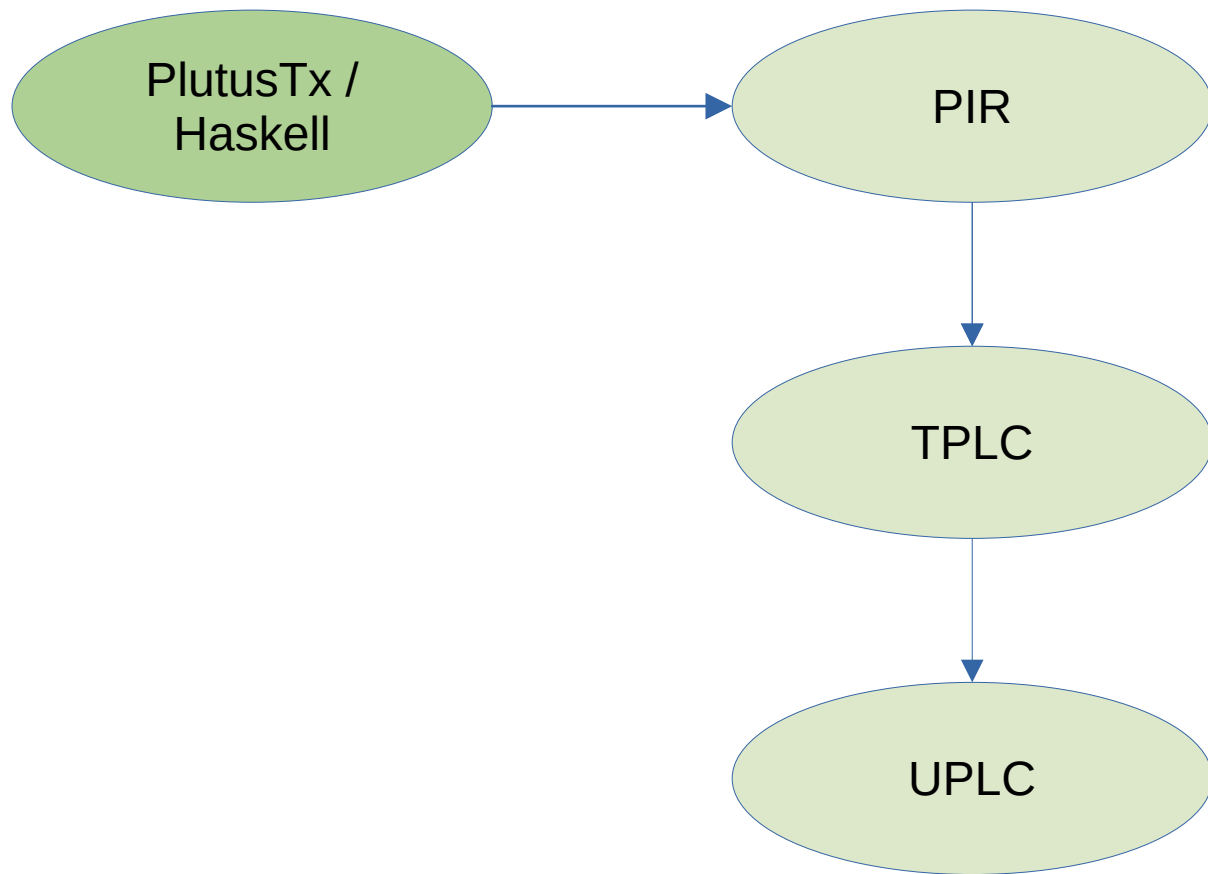
Intro: Compilation Pipeline



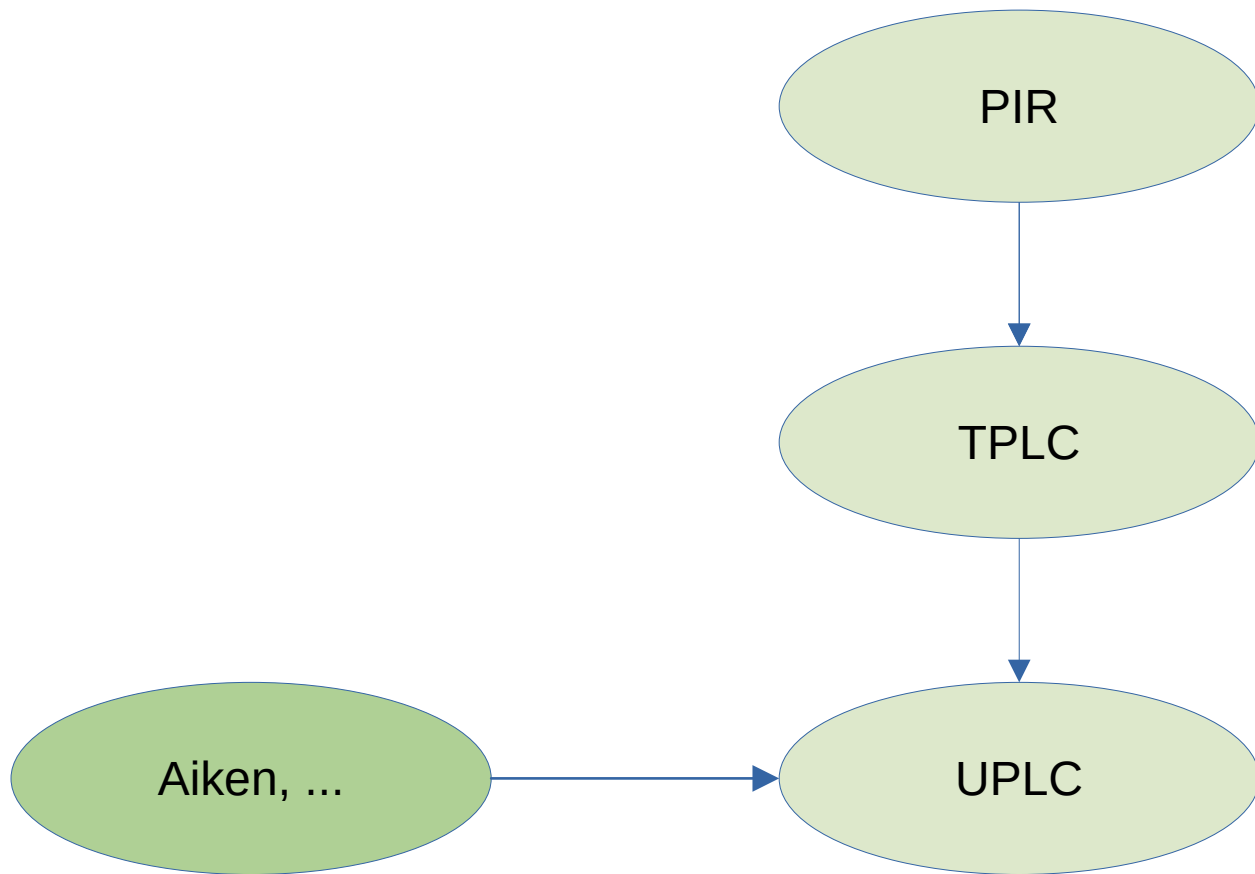
Intro: Compilation Pipeline



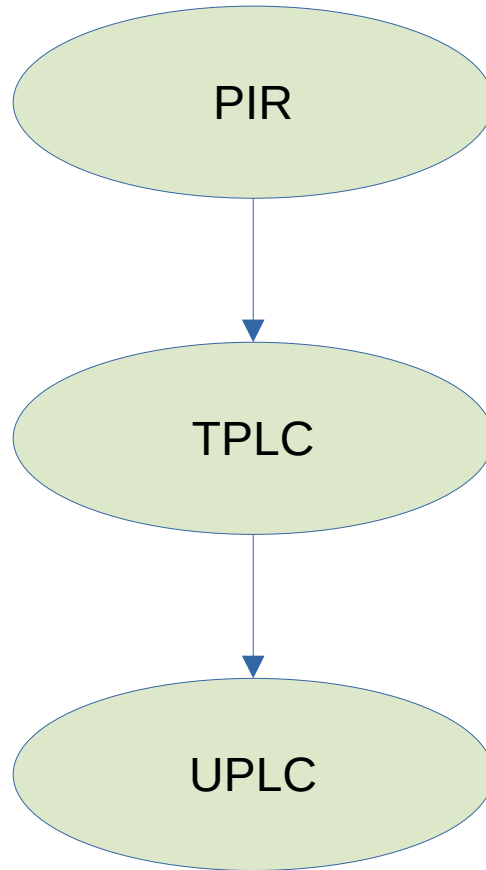
Intro: High-level Languages



Intro: High-level Languages



There is more to it...



Parameterised PIR,TPLC,UPLC

1) Over **variable names**

- `Name / DeBruijn / NamedDeBruijn`

2) Over **annotations**

- `() / Plutus-SrcSpans / Haskell-SrcSpans`

- Examples:

- ``PIR Name ()``
- ``UPLC DeBruijn SrcSpans``

Containers for PIR,TPLC,UPLC

- Plain Text
- Flat
- CBOR
- Data*
- ...

How to get script output (A)

- Programmatically inside Tx “offline” code:

- `getPlc => UPLC NamedDeBruijn SrcSpans`

- `getPir => PIR Name SrcSpans`

- `getPlcNoAnn => UPLC NamedDeBruijn ()`

- `getPirNoAnn => PIR PIR.Name ()`

, then use `System.IO.writeFile (show ...)`

How to get script output (B)

- Using a PlutusTx plugin flag:

```
{-# OPTIONS_GHC -fplugin-opt PlutusTx.Plugin:dump-pir #-}
```

```
==> Flat (Pir Name ())
```

, same with `dump-tp1c` , `dump-up1c`

How to get script output (C)

- From the *plutus-ledger-api*:

```
> import PlutusLedgerApi.Common
```

- `serialiseCompiledCode => SerialisedScript` ,i.e. **CBOR**
- `deserialiseScript => ScriptForEvaluation`
- `uncheckedDeserialiseUPLC => UPLC.Program UPLC.DeBruijn ()`

Introducing *plutus* CLI

- Unifies pir.exe, plc.exe, uplc.exe
- Features:
 - *Convert* between container formats
 - *Syntax*-check & *Type*-check plutus code
 - *Compile* to lower-level plutus code (pir → plc → uplc)
 - *Optimise* and (partial-)*apply* plutus snippets
 - *Run* or *Debug* code

How to build *plutus* CLI

```
> git clone https://github.com/IntersectMBO/plutus
```

```
> nix build .#cabalProject.x86_64-  
linux.hsPkgs.plutus-core.components.exes.plutus
```

- Without Nix (for the adventurous):

```
> cabal build plutus
```

Running plutus CLI

```
> ./plutus --help
```

```
> cabal run plutus -- --help
```

USAGE: **plutus** [FILES...] [-o FILE] [--run|--debug]...

Input/Output File extensions

.pir => PIR Name ()

.flat => Flat (UPLC NamedDeBruijn ())

.cbor => CBOR (Flat (UPLC DeBruijn ())

.data => CBOR BuiltinData

***** => UPLC Name ()

Use **-x** , **-n** , **-a** to override

ex1

```
> echo "(program 1.1.0 (con integer 1) )" > 1.uplc
```

```
> ./plutus 1.uplc
```

Try:

- convert
- --run
- --run --budget=CPU,MEM

Ex1,ex2,ex3,ex4,ex5

github.com/IntersectMBO/plutus-cli-demo

Ex5 : Spending Validator

f :: DatumType → RedeemerType → ScriptContext → Bool

f :: BuiltinData → BuiltinData → BuiltinData → ()

- Simulate validator execution

Ex5: Debugging Plutus

- Debugging plain uplc
- Injecting PlutusTx SrcSpans