LambdaBuffers questionaire

9 responses

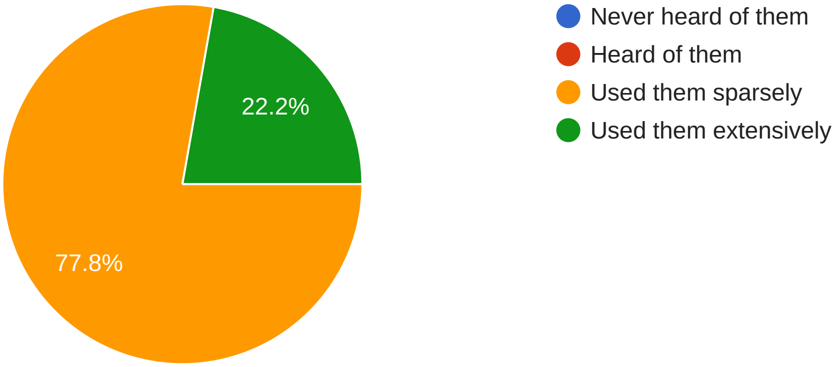
Publish analytics

Familiarity with related tech

The following describes my familiarity with technologies like Google Protocol Buffers and JSON Schemas

Copy

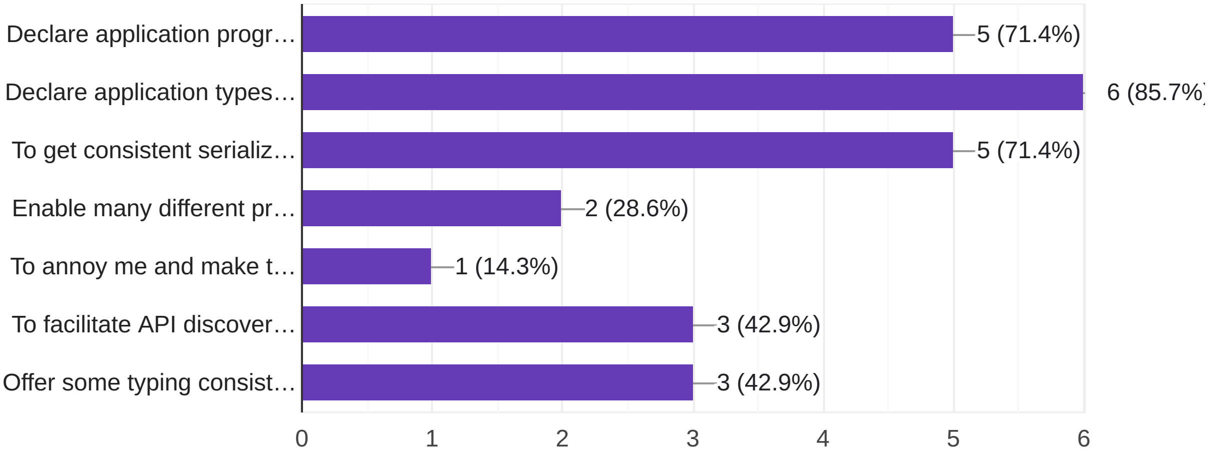
9 responses



These technologies are used to

Copy

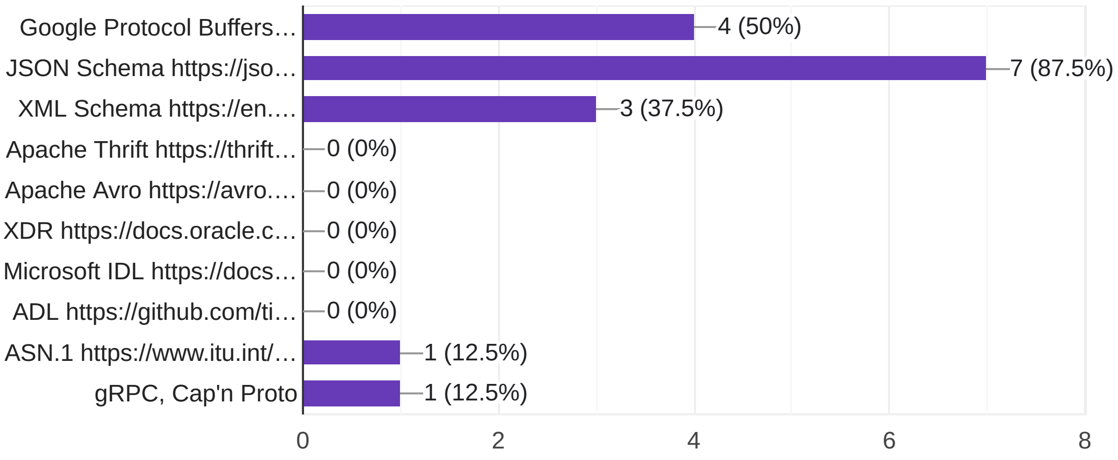
7 responses



I worked with

Copy

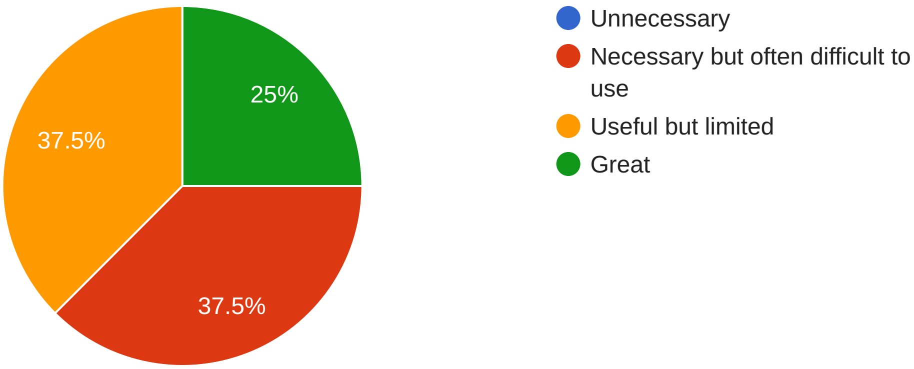
8 responses



I think these technologies are

Copy

8 responses

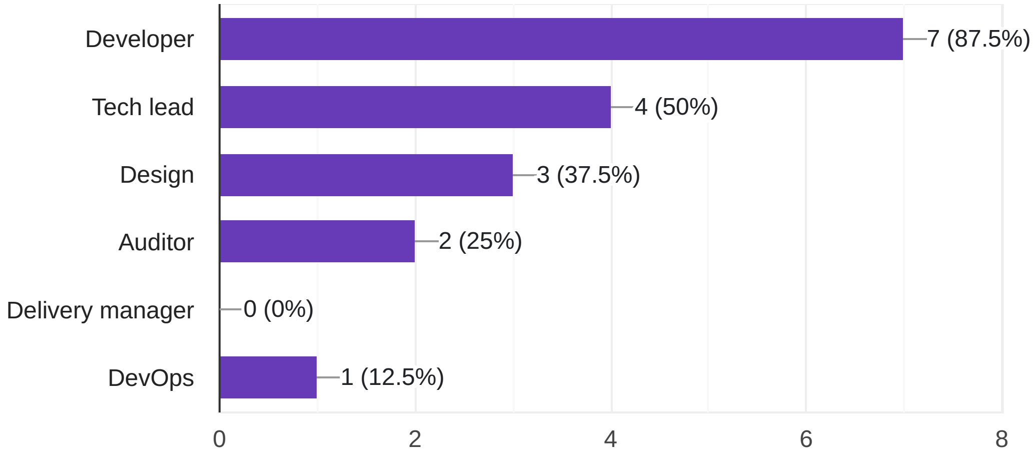


Cardano

The following describes my experiences with developing Cardano decentralized applications (dApps)

Copy

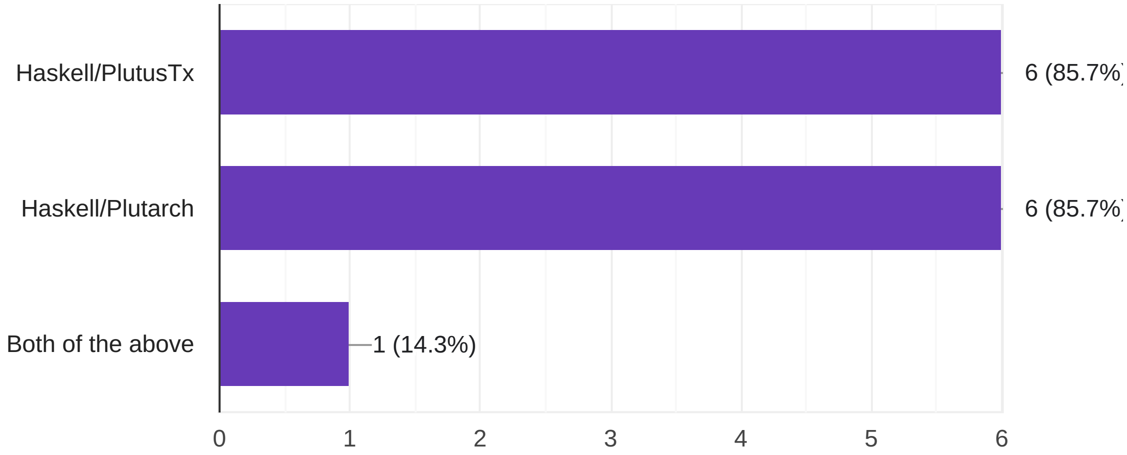
8 responses



I worked with the following languages/frameworks/libraries to specify Plutus scripts

Copy

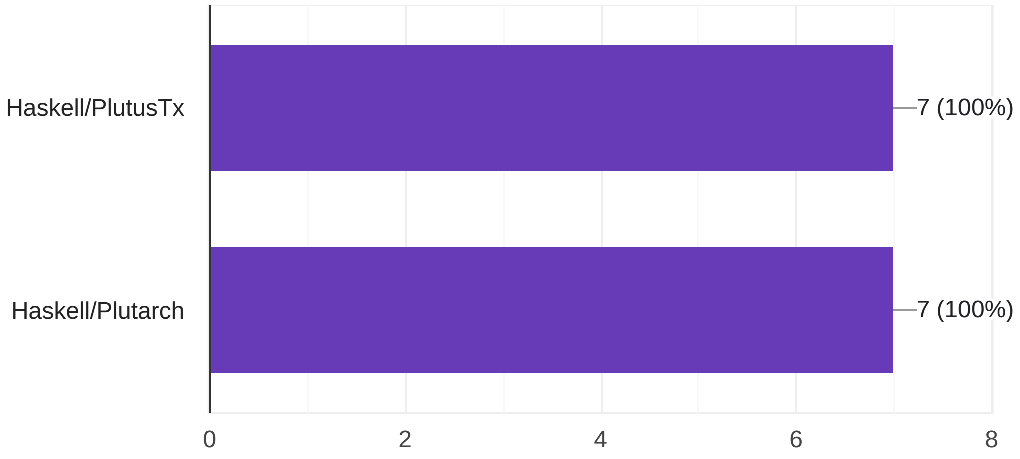
7 responses



I worked with the following languages/frameworks/libraries to specify the types for my Cardano dApp Datums (and Redeemers)

Copy

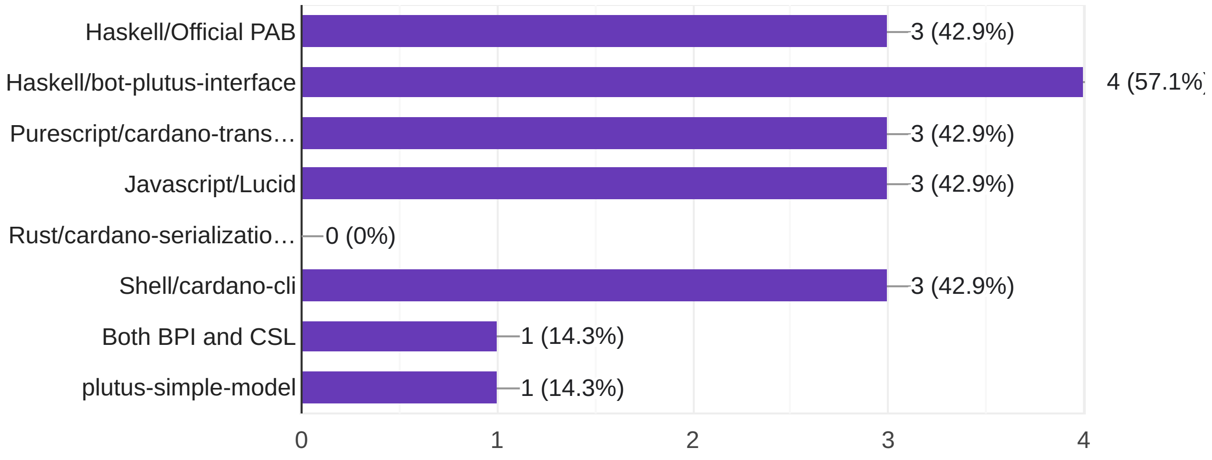
7 responses



I worked with the following languages/frameworks/libraries to build Cardano transactions for my dApp

Copy

7 responses

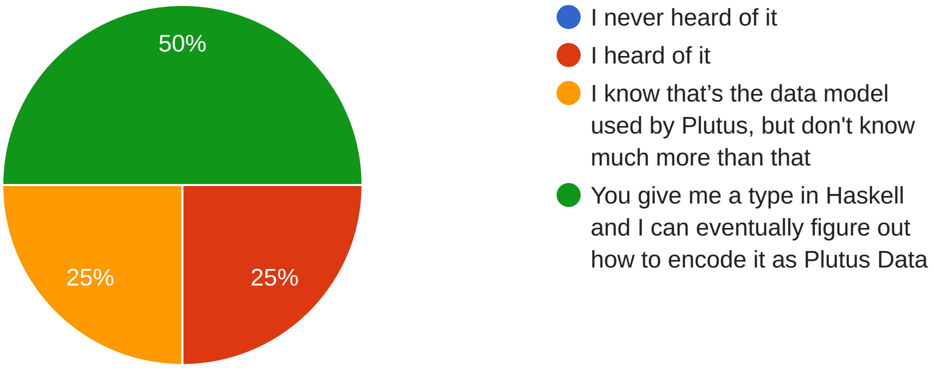


Cardano -> Plutus Data encoding

The following describes my knowledge of Plutus Data encodings:

Copy

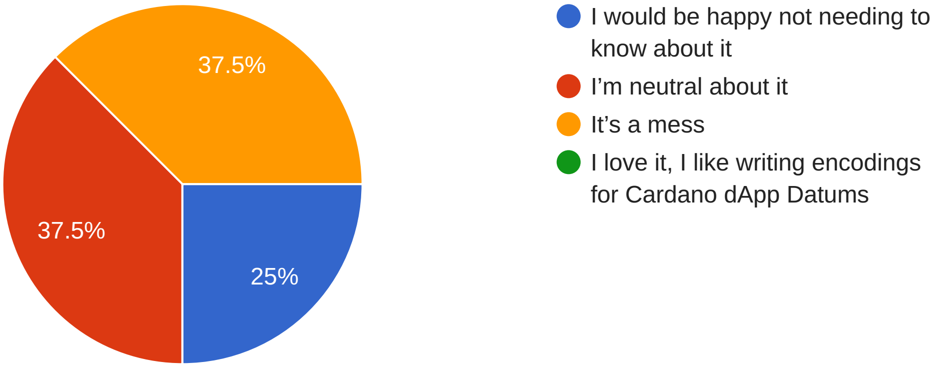
8 responses



The following describes my sentiments about Plutus Data encodings:

Copy

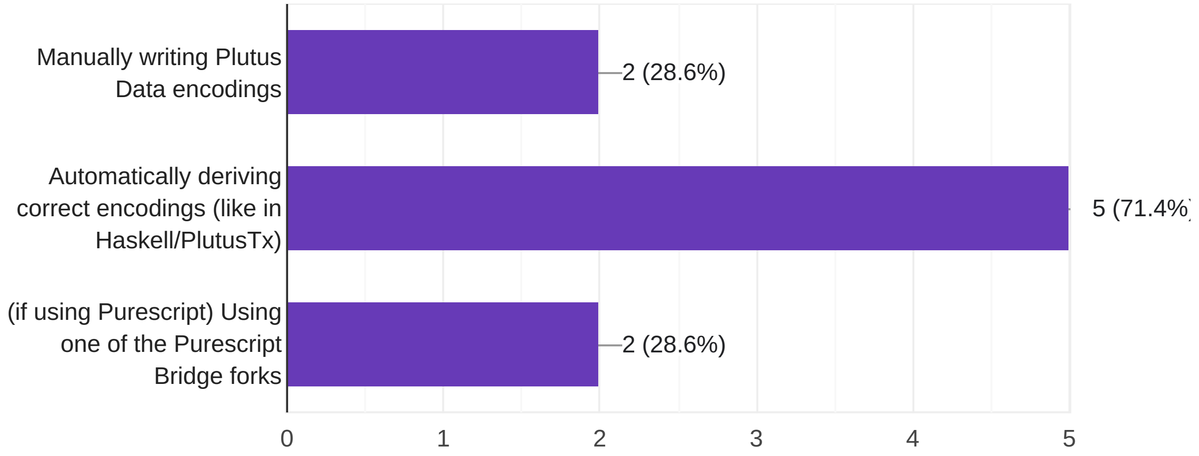
8 responses



My offchain (transaction building) dApp component manages Plutus Data encodings by:

Copy

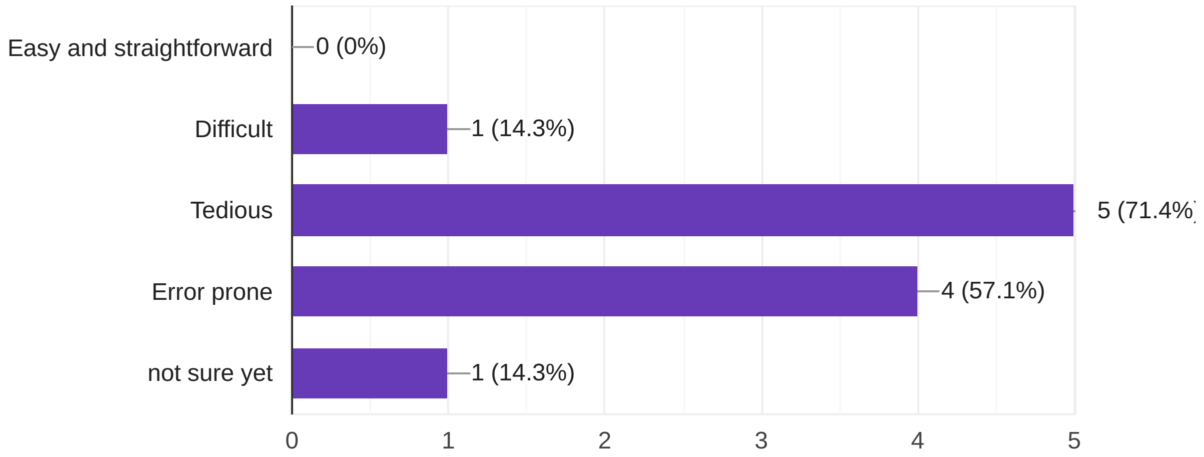
7 responses



Managing the evolution of my dApp Datum types is:

Copy

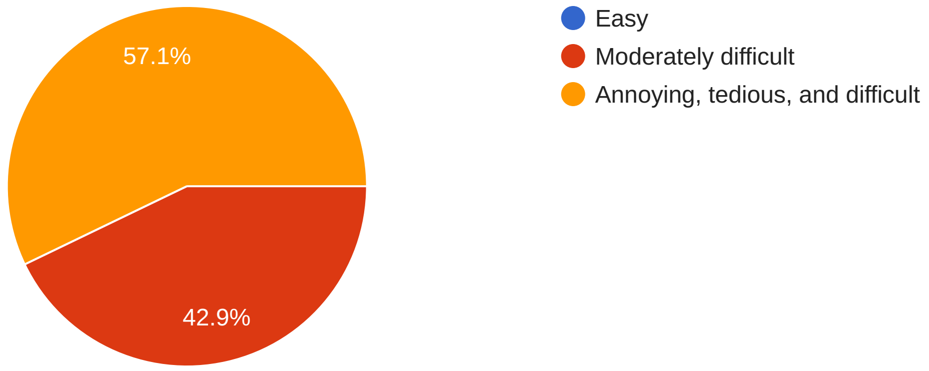
7 responses



Ensuring that Plutus Data encodings are correct when building transactions in non-Haskell environments is:

Copy

7 responses

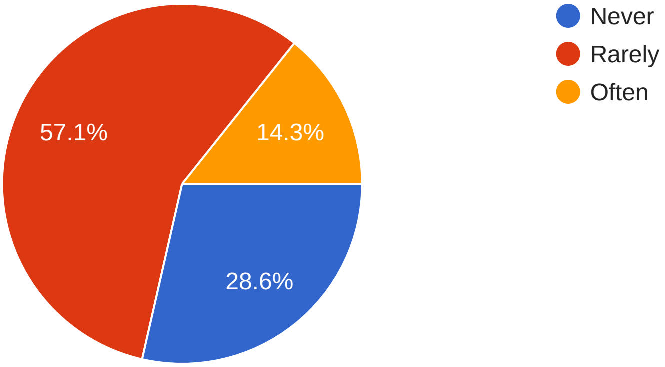


Cardano -> Configuration management

I find myself manually writing JSON/Yaml encodings for sharing dApp configurations

Copy

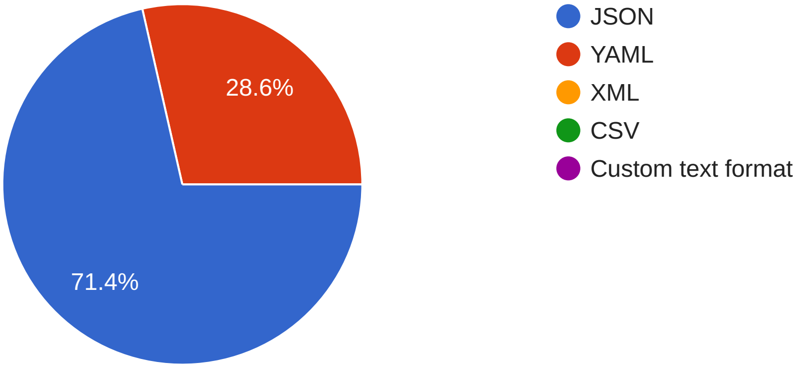
7 responses



The format I use the most is

Copy

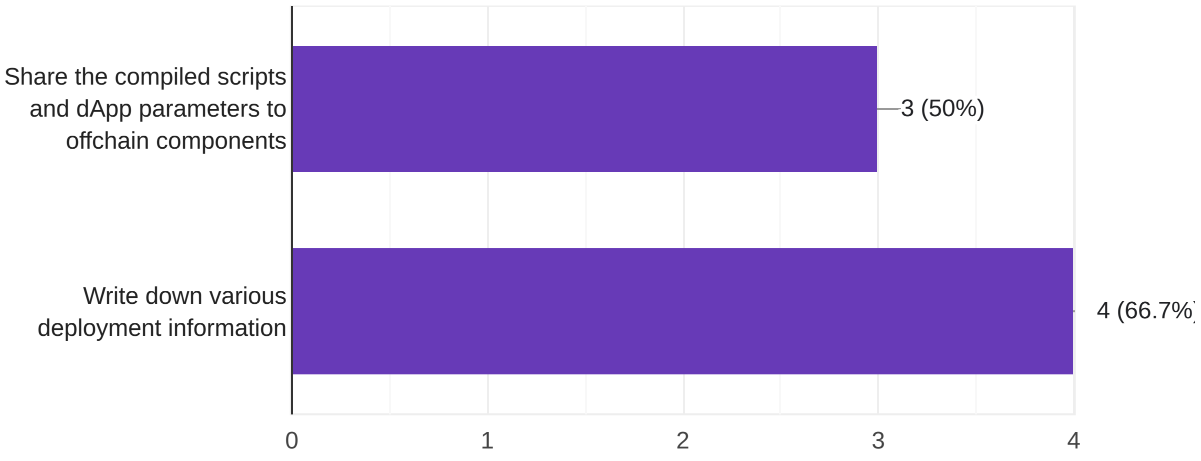
7 responses



I use ad-hoc configuration files to

Copy

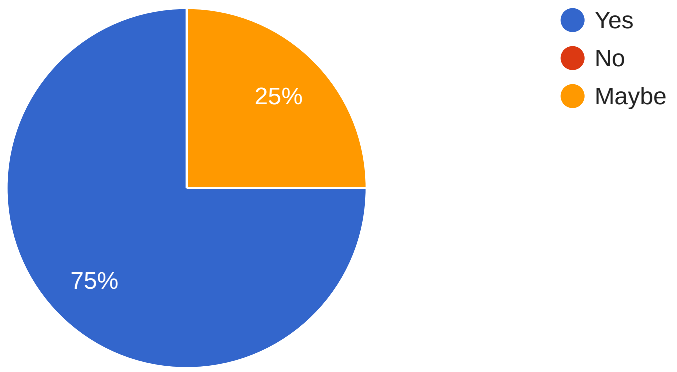
6 responses



I would like to have my configuration files specified with some Schema (like JSON Schema for example) and make reading and writing said configuration seamless across language boundaries

Copy

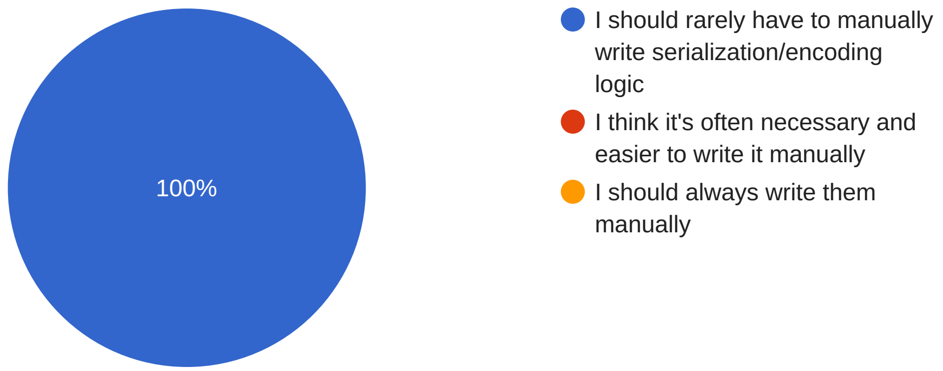
8 responses



The following describes my sentiment about manually managing various encodings and formats (coding and testing):

Copy

8 responses

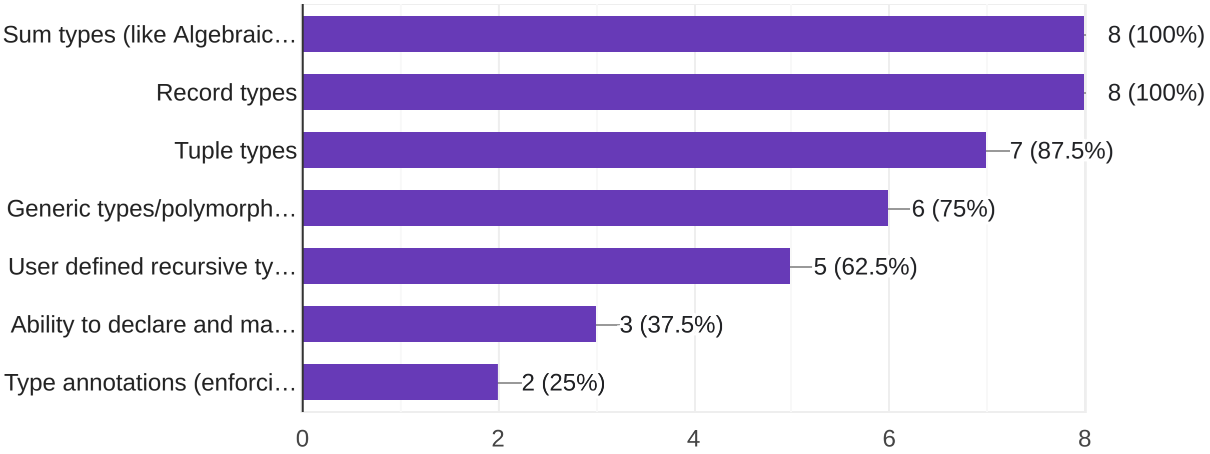


LambdaBuffers

I would like the following features in a schema language

Copy

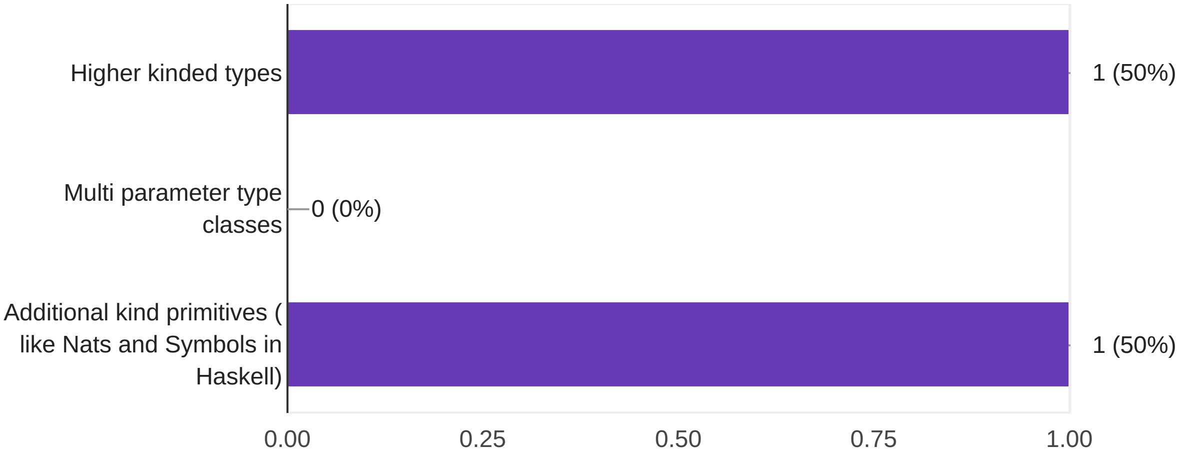
8 responses



I would be interested in some exotic features like

Copy

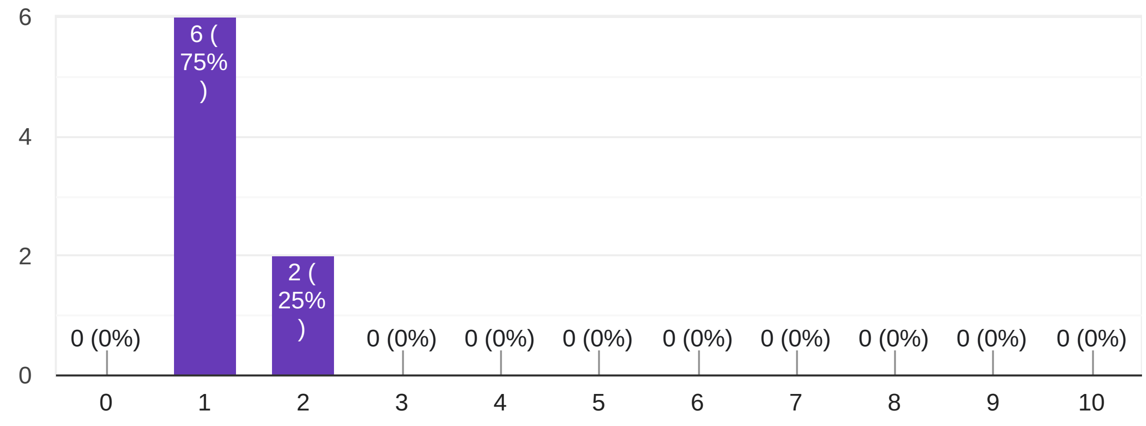
2 responses



I spend the following amount of time (x10%) on managing encodings/serialization of my application types

Copy

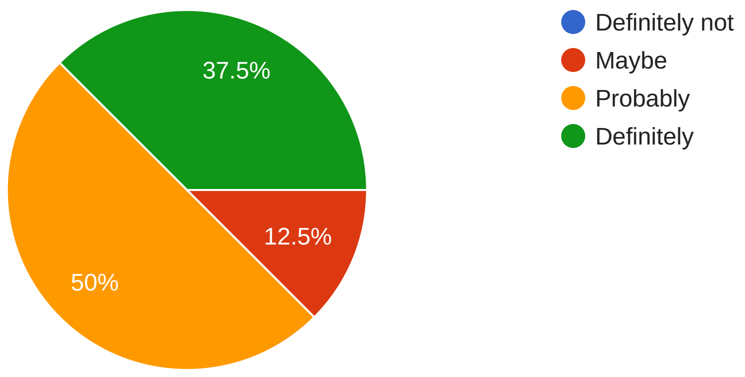
8 responses



I would use LambdaBuffers to specify and share my Cardano dApp datum types and make them seamlessly available 'for free' in supported language environments (Haskell, Rust, Purescript, Javascript etc.)

Copy

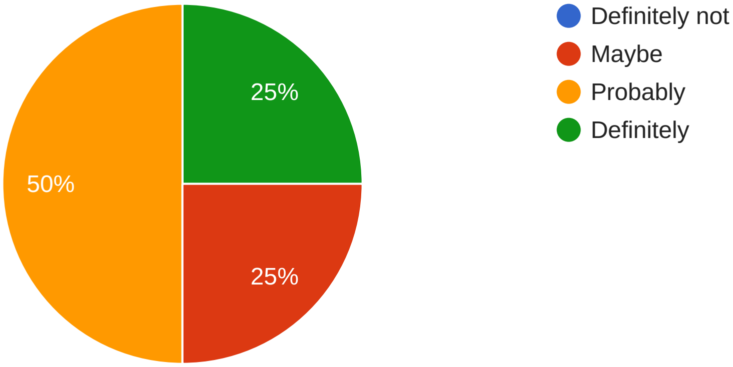
8 responses



I would use LambdaBuffers to specify and share my application API types and make them seamlessly available 'for free' in supported language environments (Haskell, Rust, Purescript, Javascript etc.)

Copy

8 responses



I want to tell you this as well....

2 responses

There are multiple ways to write Data encoding for a given type (e.g. You could serialize Bool as an integer to be efficient, but IOG decided not to. Similarly with records with only one constructor, newtypes, etc). Would be really nice to be able to specify desired encoding flavor per type

Good luck! You got this and I believe in you folks and your project!