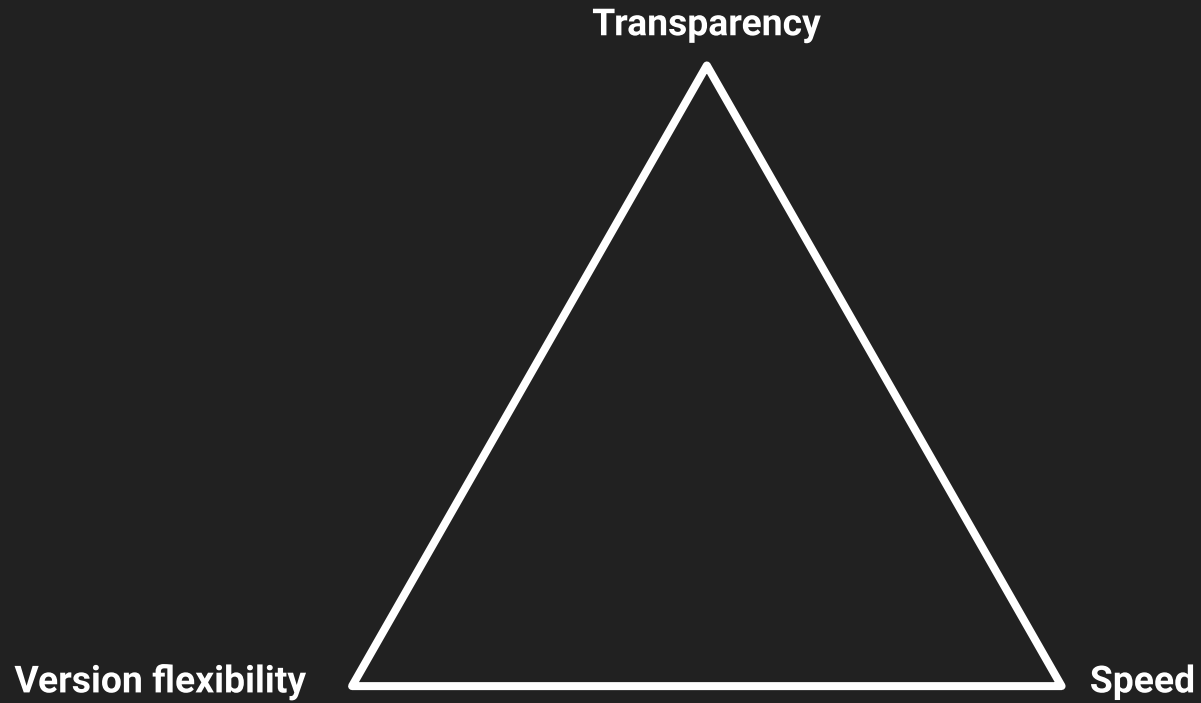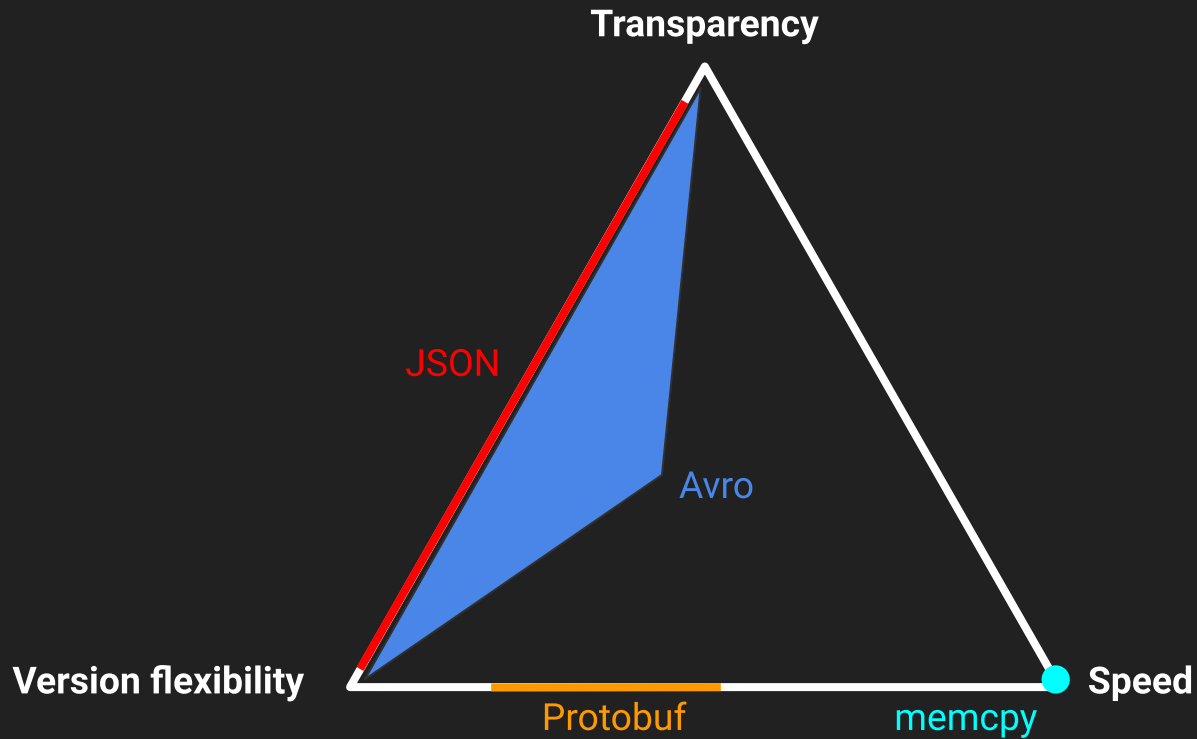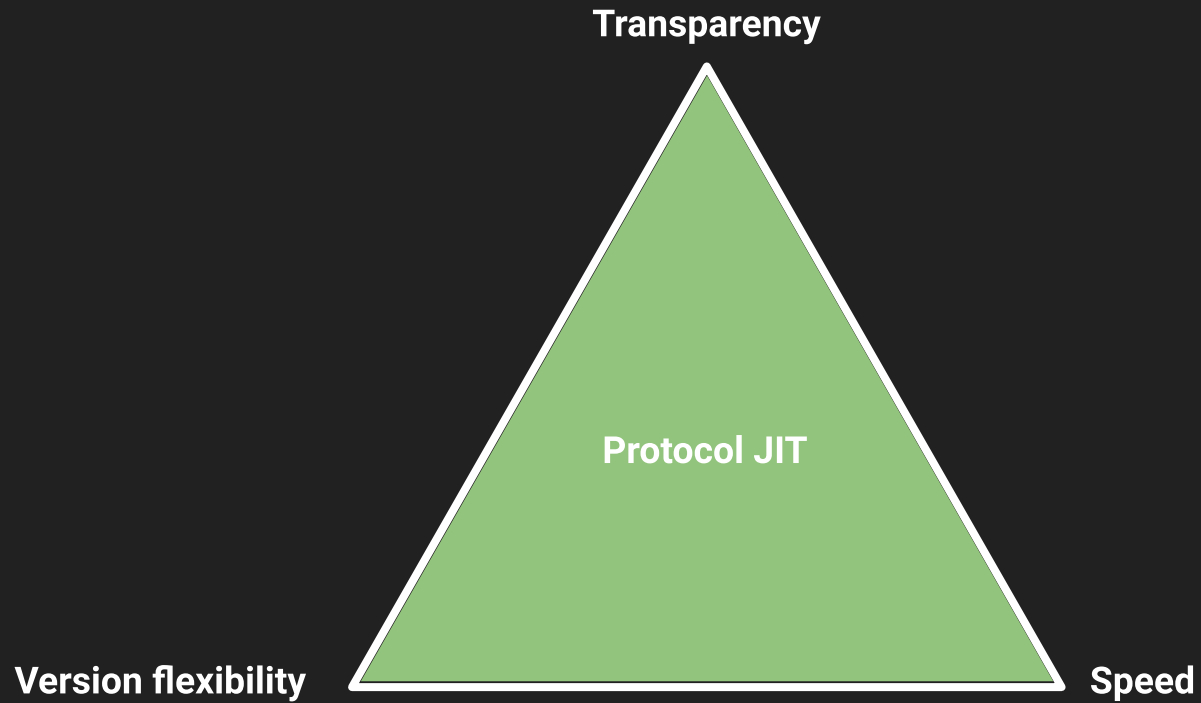# Protocol JIT

Motivation

Motivation

Motivation

Types

Integers

`intN | uintN | charN`

- Signed, unsigned and signless types with N bits
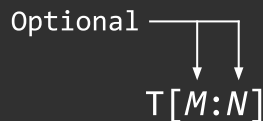- Currently supports 8, 16, 32, and 64 bits

Types

Arrays

$$T[N]$$

- Fixed length sequence of type T with exactly N elements

Types

Vectors

```
Optional ┐
         │  │
         ▼  ▼
    T[M:N]
```

- Variable-sized sequence of type T with at most N elements, or no maximum if N is unspecified
- M is a hint to the compiler that it will rarely contain more than M elements

Types

Structs

Optional

```
struct S external {
    field: T;
    ...
}
```

- Product type
- Generates a struct in C++ when compiled using **pjc**
- *external* indicates that the struct definition already exists and only supporting code is generated

Types

Variants

```
variant V {
  term: T = N;
  ...
}                        Optional
```

- Sum type
- Contains an *undef* term by default for decoding unknowns
- Generates the following struct in C++ when compiled with **pjc** (union entry omitted if T is not provided)

```
struct V {
  enum class Kind { term = N, ... } tag;
  union { T term, ... } value;
};
```

Types

Enums

```
                                  Optional
enum E {
   term = N;
   ...
}
```

- Special case of variants with no data associated with terms
- Contains an *undef* term by default for decoding unknowns
- Generates the following output in C++ when compiled with **pjc**

```
enum class E {
   term = N,
   ...
};
```

Functions

Encoding

```
                         ┌──── Optional
enum E {                 ↓
  term = N;
  ...
}
```

- Special case of variants with no data associated with terms
- Contains an *undef* term by default for decoding unknowns
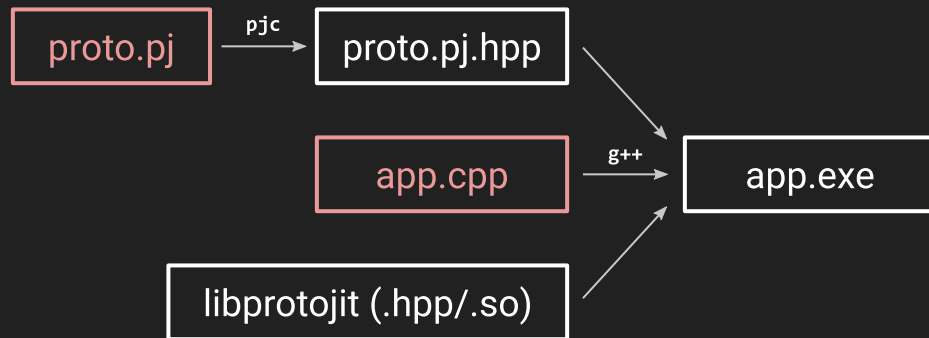- Generates the following output in C++ when compiled with **pjc**

```
enum class E {
  term = N,
  ...
};
```

Files

`*.pj`	Contains the description of the protocol, with type definitions, namespaces, and a tag for the top-level protocol type
`*.pj.hpp`	File generated by **pjc** from *.pj files with struct and variant definitions, as well as code describing the exact in-memory layout of these definitions
Schema file	Serialized description of an optimized wire protocol generated using **libprotojit.so** that other readers and publishers can use

Workflow
Compile-time

Workflow
Runtime