

Serialization Formats

One of the driving forces in developing CosmWasm, besides security by design, was to include a very nice Developer UX. Key to this is the ability to inspect and debug messages sent on the blockchain and parse results without needing complex libraries. Also not requiring downloading custom schemas and ABIs just to make a method call.

JSON

The natural solution was to use JSON everywhere. It is self-describing, human-readable, and used in APIs everywhere. It does have some downsides, such as handling numbers over 2^{53} (just use strings), no clear distinction between strings and base64-encoded binary, and no hard-coded schema. We auto-generate [JSON Schema](#) descriptors for the [public API of contracts](#), which can be used to inspect the supported API and optionally used in client-side tooling for auto-validation of messages.

The feedback when developing and debugging with this has been positive, and we are quite happy with the Developer UX with this. It is too early to tell if the message size and free-form schema will become a hindrance in production. However, please note that contracts define their own parsing logic for messages, the codec is not enforced by the framework. We provide first-class support for JSON through [cosmwasm::serde](#) and [cw-template](#), but anyone can swap this out - provided they provide client support for the format.

It is helpful to have consistency to aid client development, as well as contract-contract calls.

Protobuf

Protobuf is a well-known and widely-supported binary format. It gives a stricter schema guarantee than JSON and a more compact format. Protocol Buffers and GRPC support have been added with the Cosmos SDK v0.39.0 upgrade.

Cap'n Proto

[Cap'n Proto](#) is a super-lean encoding format with zero-copy reads, and no parsing needed. This has been [suggested for use in CosmWasm](#) as an optional addition. This may be considered as an opt-in format for contracts desiring such efficiency or strict schema, or possibly just used for encoding internal data structures (Params). [Previous Querying Next Contract Composition](#) * [JSON](#) * [Protobuf](#) * [Cap'n Proto](#)