Construct	Description	Syntax element	Type of $value
Smart contract			
✗ SCDL version	Version of SCDL used; in this paper the version is 1.0	`scdl_version : $value`	`String`
✗ Name	Expressive name of contract	`name : $value`	`String`
✗ Contract version	Version of smart contract	`version : $value`	`String`
☐ URL latest version	Optional URL to latest descriptor	`latest_URL: $value`	`URL`
☐ Description	Free text description of contract	`description : $value`	`String`
✗ Author	Developer name of contract	`author : $value`	`String`
✗ Creation date	Date the descriptor was created	`created_on : $value`	`Date`
☐ Update date	Date the descriptor was updated	`updated_on : $value`	`Date`
☐ Lifecycle	Lifecycle state of contract	`lifecycle : $value`	`"ready"	"destroyed"`
☐ SCL address	SCL address of possible smart contract gateway	`scl : $value`	`SCL address, see Section 3`
✗ Blockchain type	Name of the blockchain platform	`blockchain_type : $value`	`"ethereum"	"bitcoin"	"fabric"	"neo"`
✗ Blockchain version	Version of the blockchain platform the contract runs on	`blockchain_version : $value`	`String`
✗ Blockchain-internal address	Blockchain-internal address of smart contract, e.g., Ethereum accout	`internal_address : $value`	`String`
☐ Metadata link	Link to external metadata of contract, if available	`metadata : $value`	`URL`
☐ Code hash	SHA256 hash of contract code	`hash : $value`	`String`
✗ State	Tells if the contract maintains internal state of not	`is_stateful : $value`	`Boolean`
✗ Functions	List of functions provided by contract	`functions : [Function, Function,...]`	`Array of Function`
☐ Events	List of events generated by contract	`events : [Event, Event,...]`	`Array of Event`
Function			
✗ Name	Contract-wide unique name of function	`name : $value`	`String`
☐ Description	Free text description of function	`description : $value`	`String`
✗ Scope	Visibility of function	`scope : $value`	`"public"	"private"	"internal"	"external"`
✗ Side-effect	Tells whether the function as side-effects on state or not	`has_side_effects : $value`	`Boolean`
☐ Dispatcher	Name of dispatcher function to use for function invocation	`dispatcher : $value`	`String`
☐ Input parameters	List of input parameters	`inputs : [Parameter, Parameter,...]`	`Array of Parameter`
☐ Output parameters	List of output parameters	`outputs : [Parameter, Parameter,...]`	`Array of Parameter`
☐ Events generated	List of names of events generated by function	`events : [String, String,...]`	`Array of String`
Event			
✗ Name	Contract-wide unique event name	`name : $value`	`String`
☐ Description	Free text description of event	`description : $value`	`String`
✗ Output parameters	List of output parameters	`outputs : [Parameter, Parameter,...]`	`Array of Parameter`
Parameter			
✗ Name	Unique parameter name	`name : $value`	`String`
✗ Abstract type	Abstract, blockchain-indepdendent data type of parameter	`abs_type : $value`	`String`
☐ Native type	Contract's programming language-native data type of parameter	`nat_type : $value`	`String`
☐ Index	Tells if parameter is indexed and therefore searchable	`is_indexed : $value`	`Boolean`

✗ Mandatory element   ☐ Optional element