MLJSON

Eric Bloch & Ryan Grimm

Why JSON

The major languages play nicely

And a bunch of the minor ones…

- **ASP**
- **ActionScript**
- BlitzMax
- **C**
- **C++**
- **C#**
- Clojure
- ColdFusion
- D
- Delphi
- E
- Eiffel
- Erlang
- Fantom
- Go
- Haskell
- haXe
- **Java**
- **JavaScript**
- Lasso
- Lisp
- LotusScript
- Lua
- **Objective C**
- Objective CAML
- OpenLaszlo
- **Perl**
- **PHP**
- Pike
- PL/SQL
- pljson
- PowerShell
- Prolog
- **Python**
- Qt
- R
- Racket
- Rebol
- RPG
- **Ruby**
- Scheme
- Squeak
- Symbian
- Tcl
- Visual Basic
- Visual FoxPro

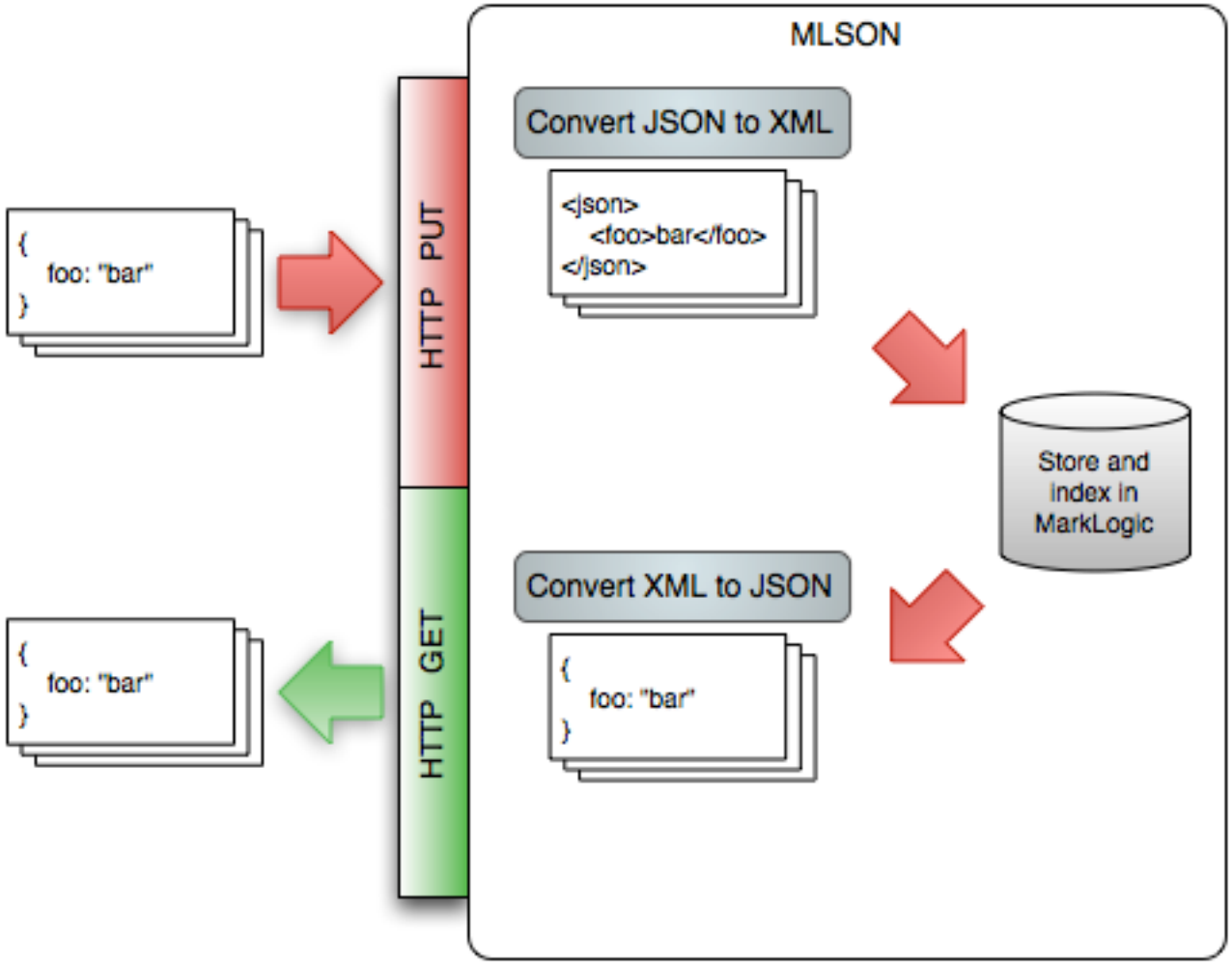
MarkLogic™

MarkLogic can store this?

```
{
    "name": "MLJSON"
    "authors": [
        "Eric Bloch",
        "Ryan Grimm"
    ],
    "version": {
        "number": 0.1,
        "isAlpha": true
    }
}
```

MarkLogic™

It can now!



MarkLogic™

Fully accessible via a REST interface

- PUT new documents
- GET existing documents
- DELETE existing documents
- POST document permissions, collections and quality
- Query for document sets by path or full text

Querying the JSON

JSON Query	XPath Evaluated
{key: “foo”}	/json[exists(foo)]
{key: “foo” value: “bar”}	/json[foo = “bar”]
{key: “foo”, value:[“bar”, “baz”]	/json[foo = (“bar”, “baz”)]
{innerKey: “foo”, value: [“bar”, “baz”]	/json[//foo = (“bar”, “baz”)]
{key: “foo”, value: {key: “id”, value: 123456}}	/json[foo/id = 123456]
{key: “price”, value: 8.99, comparison: “<“}	/json[price < 8.99]

MarkLogic™

Fully supports cts queries (full text)

```
{fulltext: {
    { contains: {
        key: "paragraph",
        string: "Hello World",
        weight: 1.0
    }}
}}
```

TO

```
cts:search(/json,
    cts:element-word-query(
        fn:QName("", "paragraph"), "Hello World", (), 1
    )
)
```

MarkLogic™

It's not all roses

Tradeoffs

- No database side programming
- No joining of documents on the server side

TODO

- Bulk updates
- Encoding of JSON keys that aren't valid QNames
- Search result snippets
- Public XQuery API for generating/manipulating JSON documents

MarkLogic™