

Introduction to XQC

The C Language Binding for XQuery

Developed as a collaboration between



Zorba (FLWOR Foundation)

<http://www.zorba-xquery.com/>



XQilla

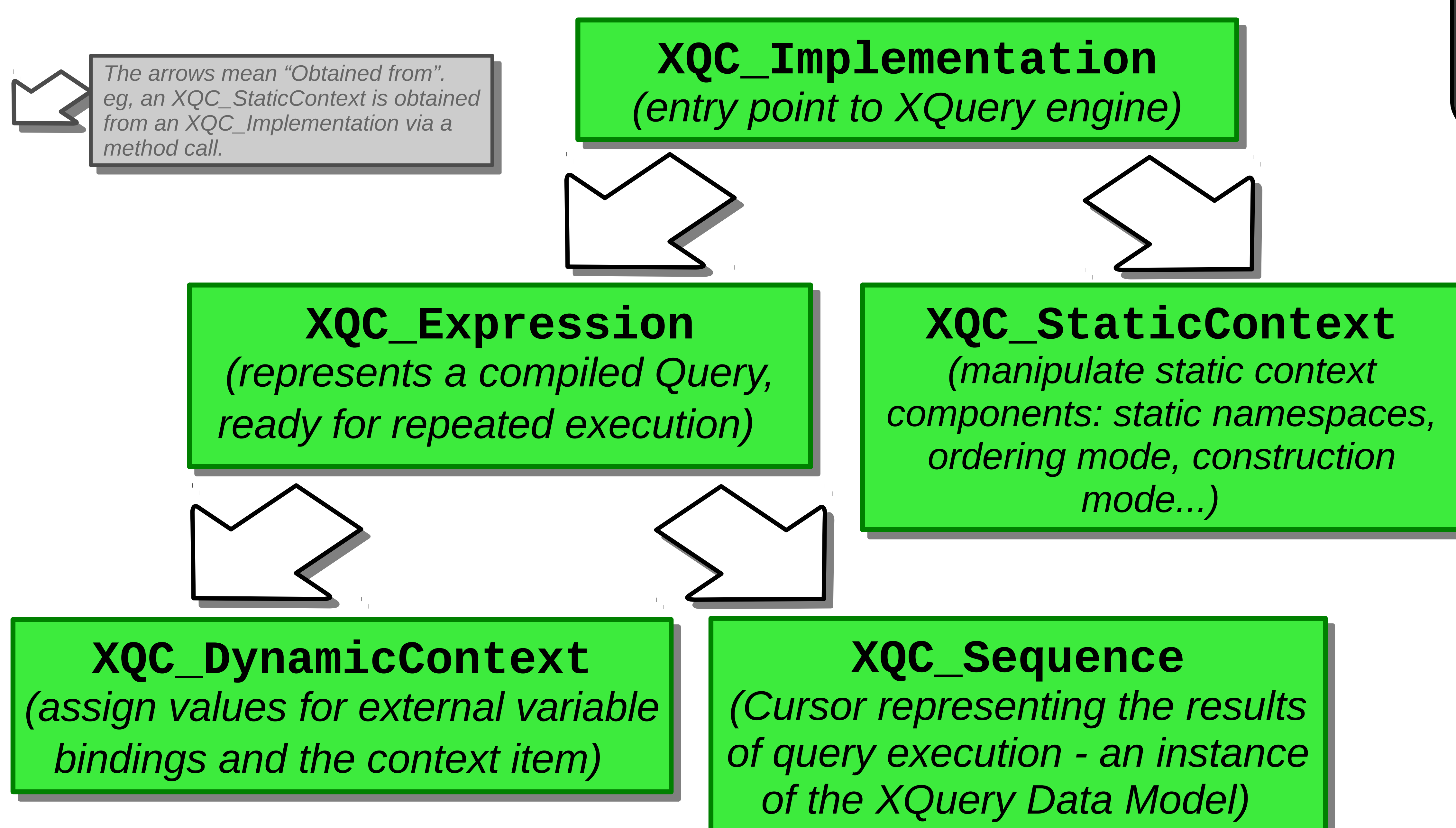
<http://xqilla.sourceforge.net/>

And already implemented by both engines!

Common API designed:

- to encourage development of tools that do not depend on a specific XQuery engine
- to allow for standardized training for XQuery/C developers
- to get more people using XQuery!

Please get involved! <http://xqc.sourceforge.net/>



Main XQC Classes and Relationships

Other XQC Features:

- Callbacks for handling errors
- Carefully documented memory management and thread support
- Utility methods for binding C values to external variables
- Extension interfaces for implementation-specific features

Annotated Code Example

```
XQC_StaticContext* stat = NULL;
XQC_DynamicContext* dyn = NULL;
XQC_Expression* query1 = NULL, query2 = NULL;
XQC_Sequence* result1 = NULL, result2 = NULL;
const char* str;

impl->create_context(impl, &stat);
stat->set_base_uri(stat, "http://example.com/");

impl->prepare(impl,
    "fn:resolve-uri(\"index.html\")",
    stat, &query1);
query1->execute(query1, NULL, &result1);

impl->prepare(impl,
    "declare variable $uri external;
    fn:concat($uri, \"#anchor\")",
    NULL, &query2);

query2->create_context(query2, &dyn);
dyn->set_variable(dyn, "", "uri", result1);
query2->execute(query2, dyn, &result2);

result2->next(result2);
result2->string_value(result2, &str);
printf("%s", str);
```

Create a **Static Context** and assign base URI for query evaluation

Compile a Query using this **Static Context**, and Execute it, obtaining **Results**

Compile a second Query

Create a **Dynamic Context** for evaluating this Query

Use **Dynamic Context** to bind External Variable "uri" to **first Query's results**

Execute Query using this **Dynamic Context**

Retrieve the string-value of first Item in Result

<http://example.com/index.html#anchor>

The Future of XQC

- Expand to support C++
- More functionality and access to XQuery features
- **Most importantly, encourage ALL C / C++-based XQuery engines to provide an XQC interface!**