In the small world of Free Software for Automation...
Beremiz is an open source IEC-61131-3 and PLCopen programming framework.
It all began in 2005 as a simple PLCopen Editor.
Today, Beremiz is an IDE for automation, and its Runtime(s)
IDE
is made of:

Runtime
IDE
An Editor,

Runtime

and a Toolchain.
Python and XSLT (GPL) Platform independent.
Model-View-Controller design
Official PLCopen's TC6 XML schema is used as Meta-Model.
Natively save & load valid TC6 XML

PLCopen XML
EXTENDED MARKUP LANGUAGE XML
Controller

Exports
IEC-61131-3
graphical
languages
as their
textual
equivalents
When building PLC, ST, IL and SFC code are first intermediate languages.
Textual
IEC-61131-3

**matIEC compiler**

Then, compiler (GPL) parse and analyse PLC code ...
Textual IEC-61131-3

matIEC compiler

C code

... and generate equivalent C code
Textual IEC-61131-3

matLEC compiler

C code

.C

Finally, C code is compiled into native target binary code.
IDE

Editor

Runtime exists in two flavors:

Toolchain

Textual
IEC-61131-3

matIEC compiler

C code

.C

GCC
Beremiz runtimes

Free Python runtime (LGPL).
C runtime for microcontrollers.
Runtime loads PLC binary.
IDE controls and monitors PLC execution remotely through RPC channel.
I/Os are done by libraries, accessing data processed by PLC.
Libraries are provided by extensions.
Extensions can do lot more than just providing libraries...
Extensions can do fieldbus

CANFestival stack (LGPL) and configurator (GPL).

EtherLAB/RTnet/Xenomai with complete configurator. Not yet disclosed publicly.
Extensions can do libraries

PLCopen TC2 Motion Control Library (proprietary).
Multi axis coordinated motion. Supports CiA402.
Extensions can do C code

```c
TXT := 'THIS IS ST CODE';
{
    printf("This is C code");
}
```

matlEC supports C code pragmas in ST.
Extensions can do C code

```
TXT := 'THIS IS ST CODE';
{
    printf("This is C code");
}
```

matIEC supports C code pragmas in ST.
User can include additional C files with PLC code.
Extensions can do Python

With Python runtime, Python extension enable asynchronous Python scripting in PLC.

Python files can also be imported. Variables can be exchanged with PLC globals.
Extensions can do HMI

WxGlade and WxPython based GUI seamlessly integrate with Python PLC runtime.
Extensions can integrate

Integrated with Smarteh's LPC Composer since 2009. Supports C runtime, CANopen, Modbus, IO modules.
Extensions can do ... Anything.
Thanks for your attention

Copyright: Edouard Tisserant

[Creative Commons license information]

Except where otherwise noted, this work is licensed under http://creativecommons.org/licenses/by-sa/3.0