

The **SAPPEUR** Language

Dipl.-Ing. (BA) Frank Gerlach (frankgerlach74@web.de)

SAPPEUR Design Objectives

A programming language that enables developers to create reliable, safe, small and fast programs for multicore processors.

Language Comparison

C++	efficient	multi-thread-unsafe
Ada	efficient	multi-thread-unsafe
.Net	inefficient	multi-thread-safe
Java	inefficient	multi-thread-safe
SAPPEUR	efficient	multi-thread-safe

SAPPEUR Objectives

support multicore microprocessors well

multi-thread safe *and* efficient

seamless integration with C++

Multi-platform

Real-time capability

Re-use existing code generators like gcc, vc++

Key SAPPEUR Features

objects are reference-counted

(all pointers are smart pointers)

all pointers either NULL or valid

synchronous destructors as in C++

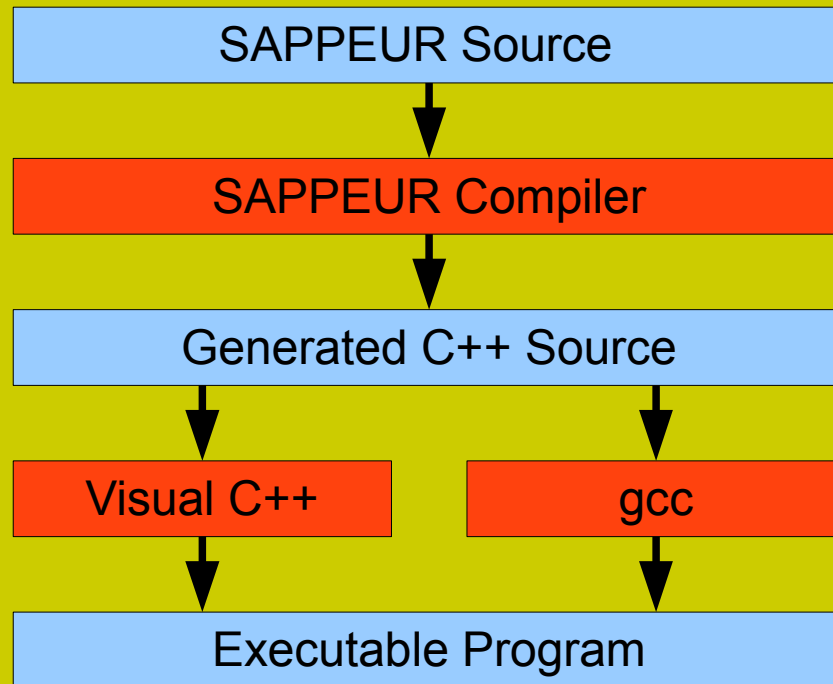
objects and arrays can be allocated on the stack
(very efficient)

type system separates single- from multithreaded
code

array bounds are always checked

no funny casting

SAPPEUR Architecture



Sample 1

SAPPEUR Code

```
Scanner::Scanner(&char str[])
{
    _currentPtr=0;
    var int i=0;
    while(str[i] != '\0')
    {
        _buffer[i]=str[i];
        i++;
    }
    _buffer[i]='\0';
}
```

Generated C++ Code

```
Scanner::Scanner(SPRStackArray<char>& str){
    _currentPtr=0;
    int i=0;
    int SPR_intScratch52;
    SPR_intScratch52=i;
    if(str._sz <= SPR_intScratch52)throw "index out of bounds";
    if(SPR_intScratch52<0)throw "index out of bounds";
    while(str._array[SPR_intScratch52]!='\0'){
        int SPR_intScratch53;
        SPR_intScratch53=i;
        if(_buffer._sz <= SPR_intScratch53)throw "index out of bounds";
        if(SPR_intScratch53<0)throw "index out of bounds";
        int SPR_intScratch54;
        SPR_intScratch54=i;
        if(str._sz <= SPR_intScratch54)throw "index out of bounds";
        if(SPR_intScratch54<0)throw "index out of bounds";
        _buffer._array[SPR_intScratch53]=str._array[SPR_intScratch54];
        i++;
        SPR_intScratch52=i;
        if(str._sz <= SPR_intScratch52)throw "index out of bounds";
        if(SPR_intScratch52<0)throw "index out of bounds";
    };
    int SPR_intScratch55;
    SPR_intScratch55=i;
    if(_buffer._sz <= SPR_intScratch55)throw "index out of bounds";
    if(SPR_intScratch55<0)throw "index out of bounds";
    _buffer._array[SPR_intScratch55]='\0';
} //End Of Method
```



Multithreading

Type System enforces thread safety of execution environment (heap, stack)

Keyword `multithreaded`

Compiler generates thread locking code

Compiler enforces single-threaded access of single-threaded classes

C++ Integration

Keyword `inline_cpp` [[<C++ code>]] copies
c++ code verbatim into generated C++ code

```
SAPPEUR void Auto::printNumber(int num)
{
    inline_cpp[[printf("%i\n", num);]]
}
```

```
Generated void Auto::printNumber(int num)
C++      {
        printf("%i\n", num);
      }
```

Not Yet Tired ?

Generics

Synchronous Destructors

Reference Counting is real-time capable (as opposed to garbage collection)

Safe Execution of Untrusted Code (“Applets”)

This slide and the next document an idea of how SAPPEUR could be used to safely execute untrusted Applets (similar to Java Applets) on a PC. Contrary to Java Applets, these would start and execute at the speed of a native application. The same applies for memory requirements.

As the SAPPEUR compiler guarantees proper execution (no array overruns, no bad pointers, no heap corruption, no stack corruption, basic multi-thread safety), Security Depends only on running the compiler chain properly.
(note that the unsafe `inline_cpp[[. .]]` is only available to core library programmers)
Certificate Authorities (CAs) like Verisign or Thawte could run the SAPPEUR compiler and the Compiler Chain “behind” and sign the resulting *exe with their Private Key.
End user browsers would validate their *exe against those CAs.
Please see the next slide for a graphic representation of this scheme.

Access to the executing PC's resources would only be through a trusted system library which enforces access restrictions on files, sockets, keyboard, user interface etc.

Signed SAPPEUR Applets

