TyCL

The TyCL compiler internals and preliminary performance analysis

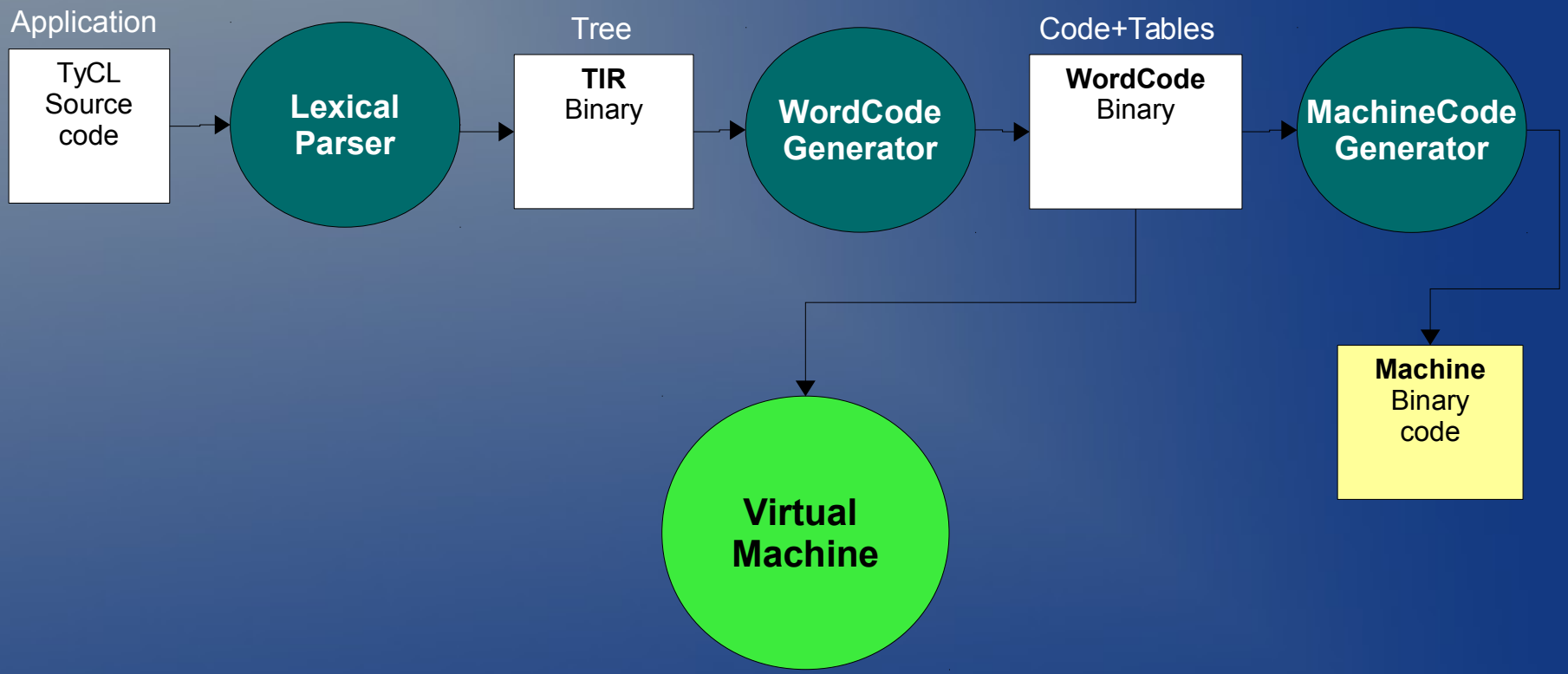
Andres Buss

Otlet Technologies

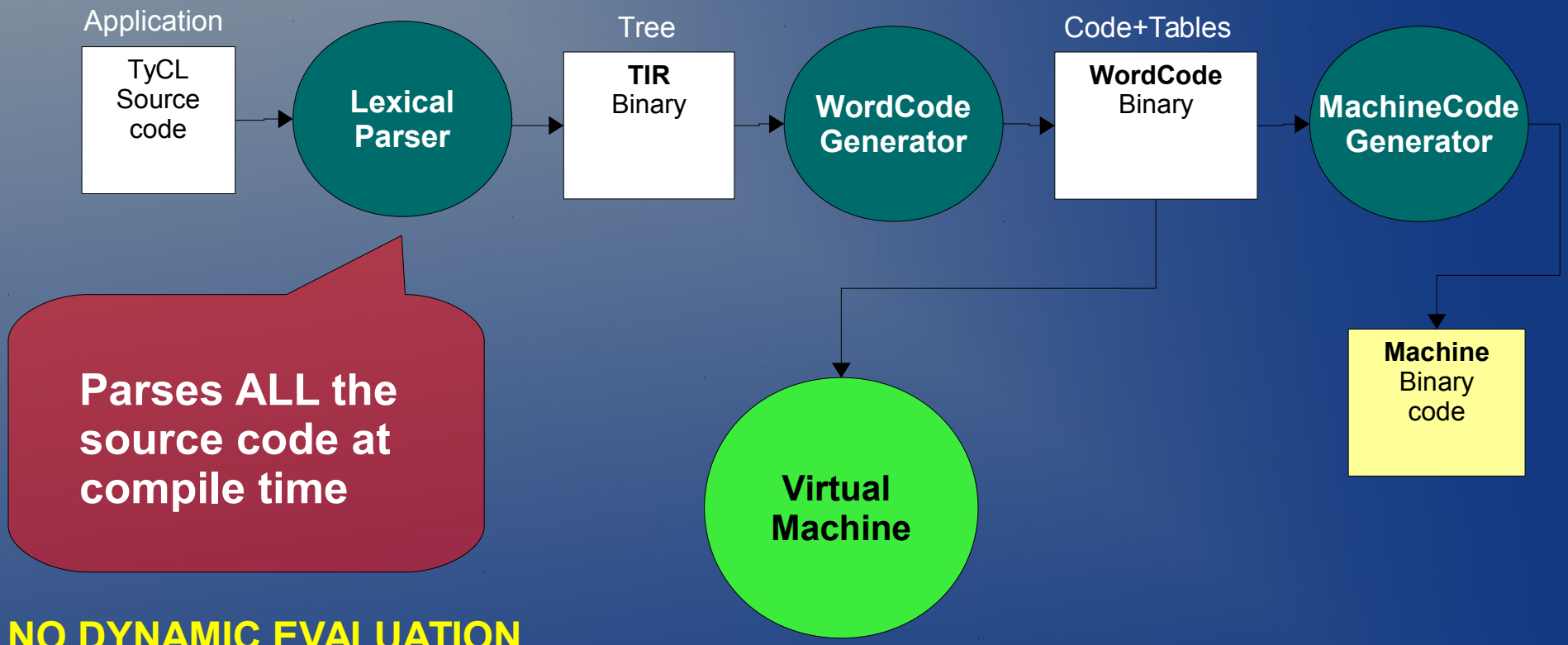
Does it really work?

How ?

Architecture

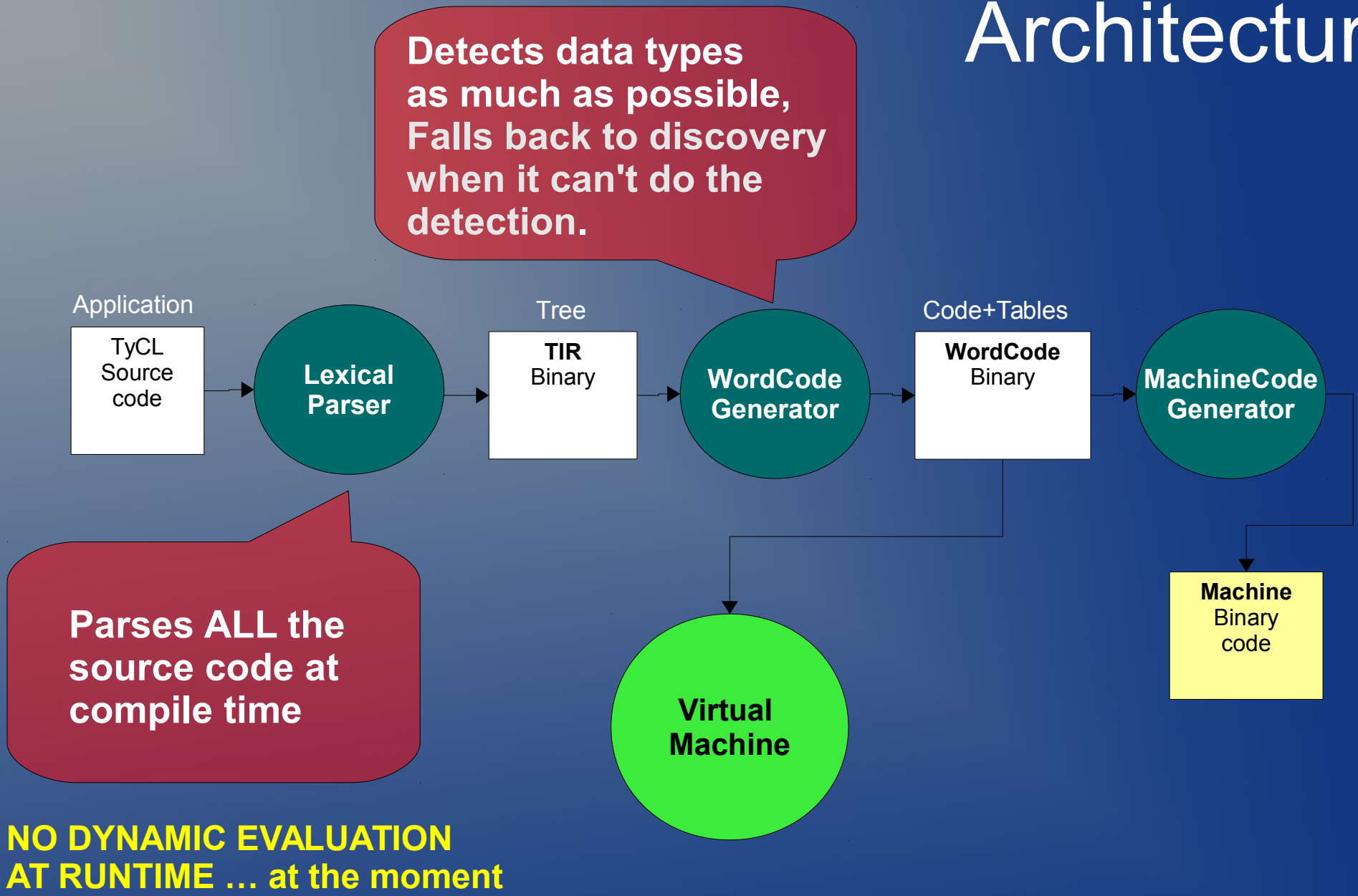


Architecture

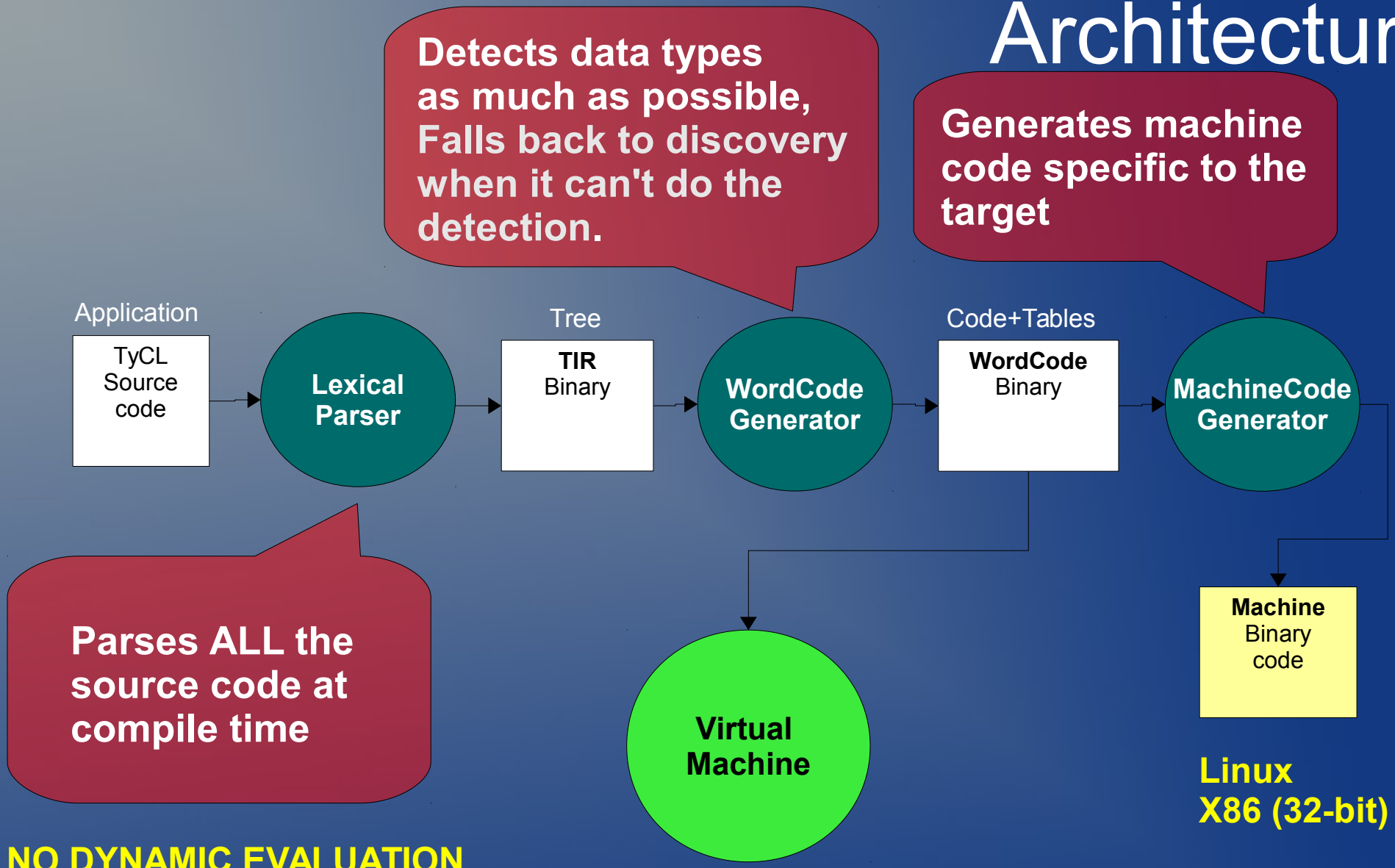


NO DYNAMIC EVALUATION AT RUNTIME … at the moment

Architecture



Architecture



The memory models

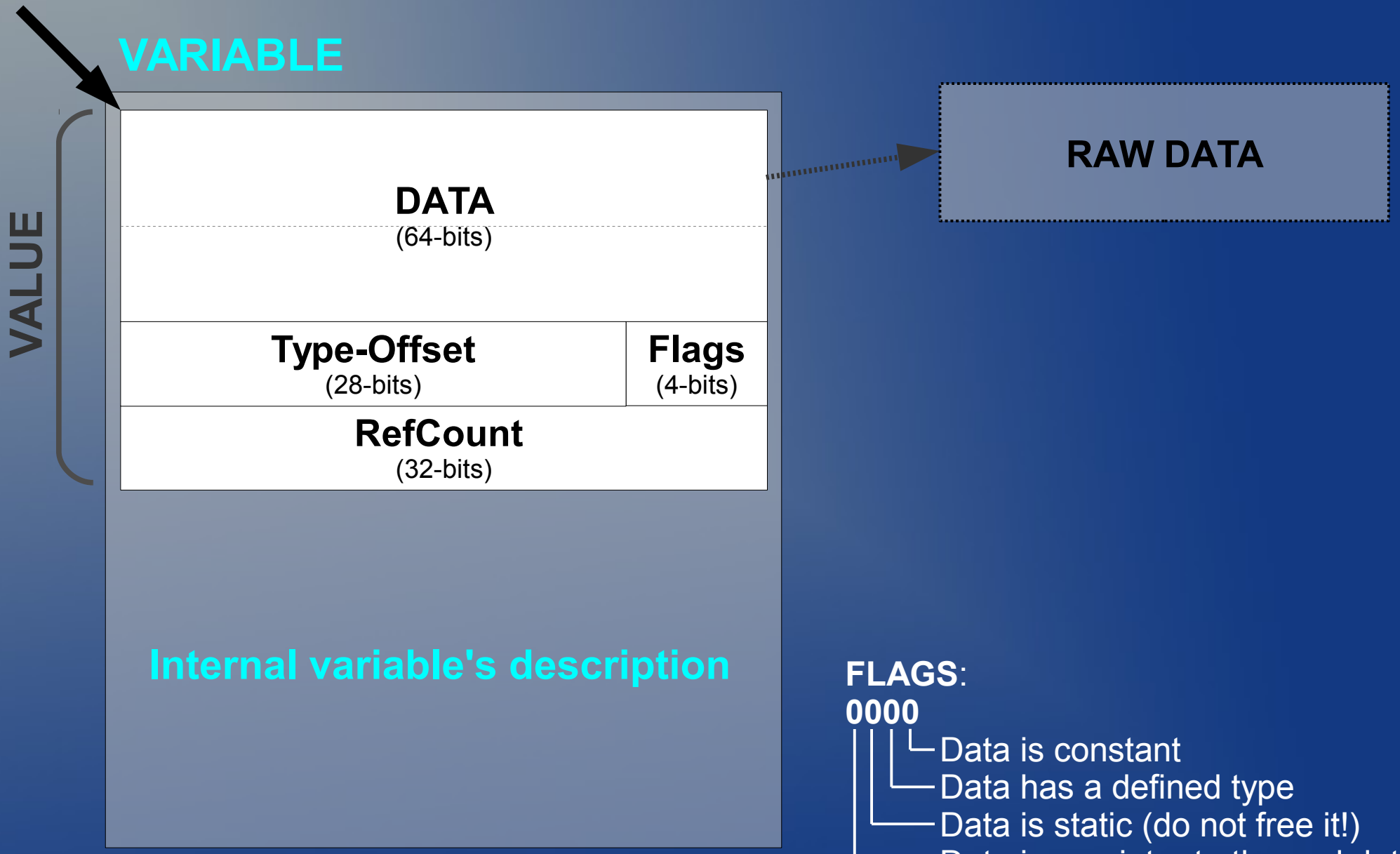
1. How are values stored in memory ?
2. How are types stored in memory ?
3. How are types related to values ?

The memory models

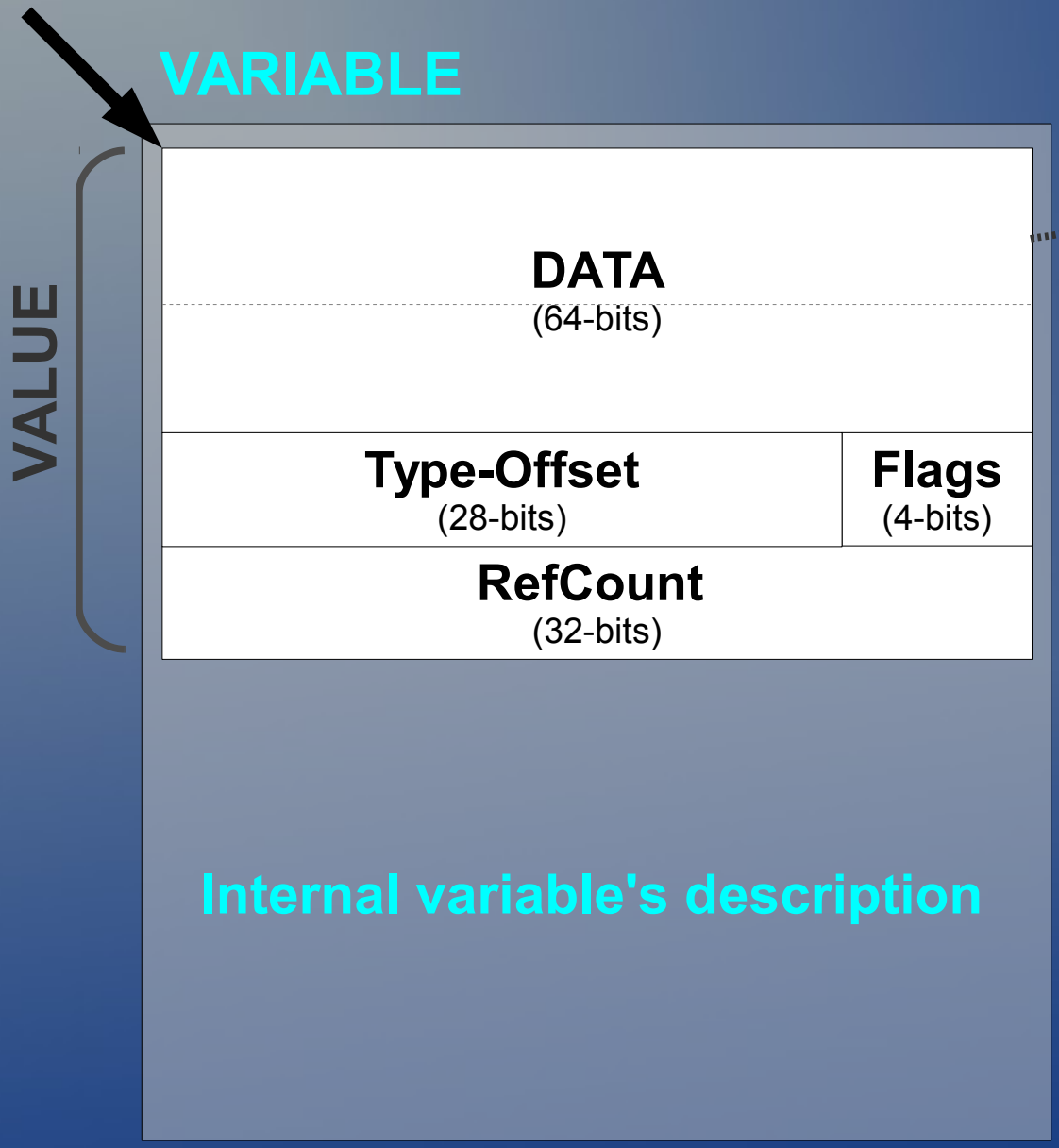
1. How are values stored in memory ?
2. How are types stored in memory ?
3. How are types related to values ?

Avoid asking / releasing dynamic memory as much as possible

Values in memory



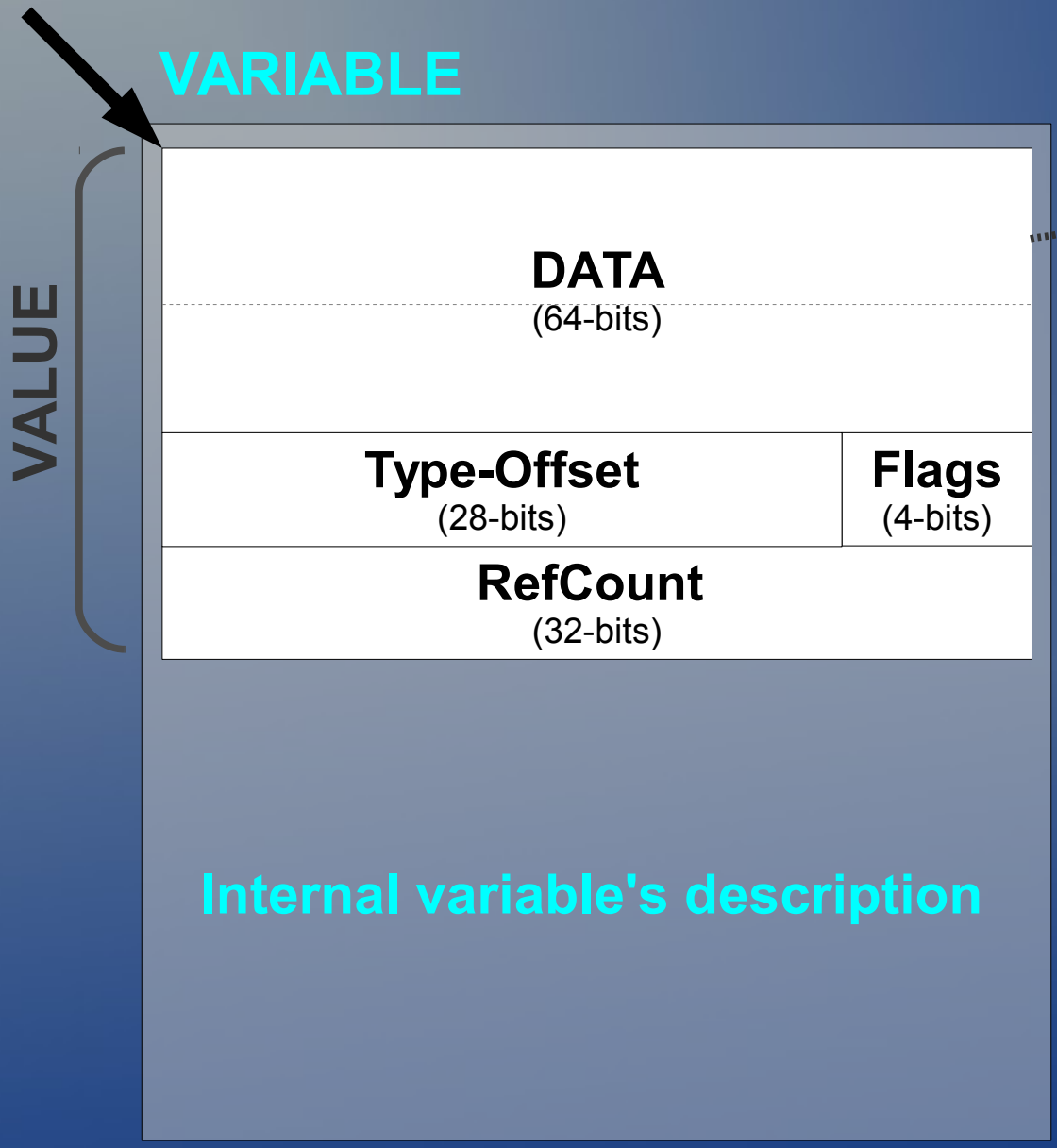
Values in memory



NO copy-on-write

FLAGS:
0000
- Data is constant
- Data has a defined type
- Data is static (do not free it!)
- Data is a pointer to the real data

Values in memory



NO copy-on-write

Always provide a place to store data

FLAGS:
0000
- Data is constant
- Data has a defined type
- Data is static (do not free it!)
- Data is a pointer to the real data

The Type system

What is a type?

struct:{

integer:flags
@tonum
@getnum
@tostr
@tobool
@set
@get
@cast
@length *@size*
@exec
@setIndex *@getIndex* *@refIndex*
@setRange *@getRange* *@refRange*
@setMember *@getMember* *@refMember*
@newframe
@namedarg
@setflag
@parmoff
@exec

}

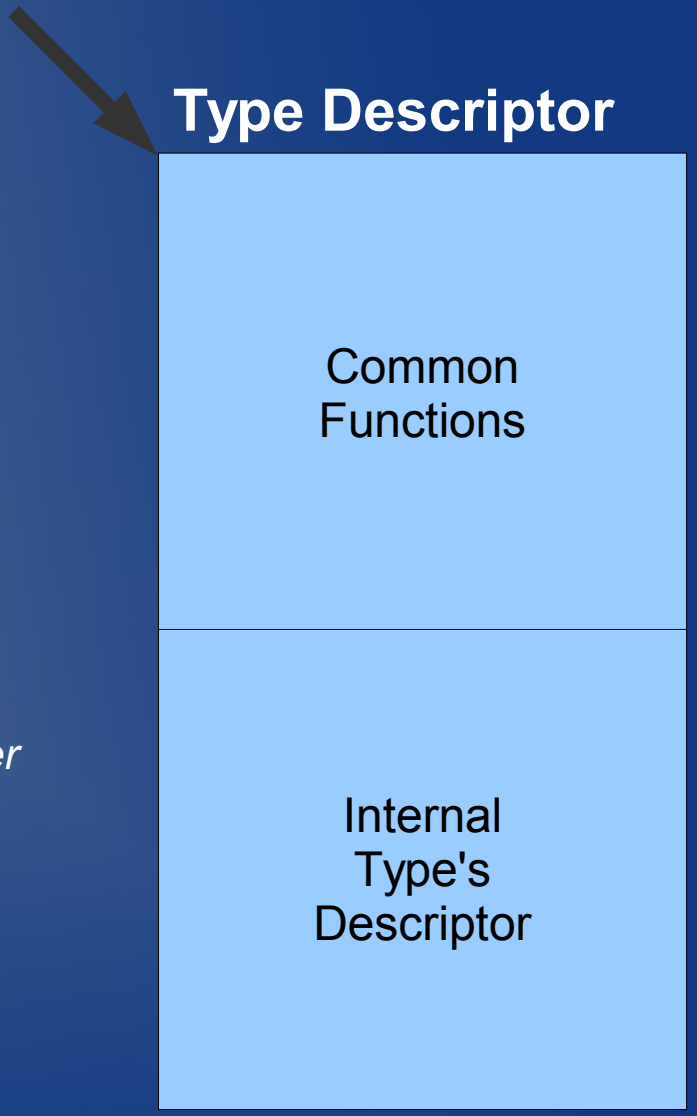
The Type system

What is a type?

struct:{

integer:flags
@tonum
@getnum
@tostr
@tobool
@set
@get
@cast
@length *@size*
@exec
@setIndex *@getIndex* *@refIndex*
@setRange *@getRange* *@refRange*
@setMember *@getMember* *@refMember*
@newframe
@namedarg
@setflag
@parmoff
@exec

}



The Type system

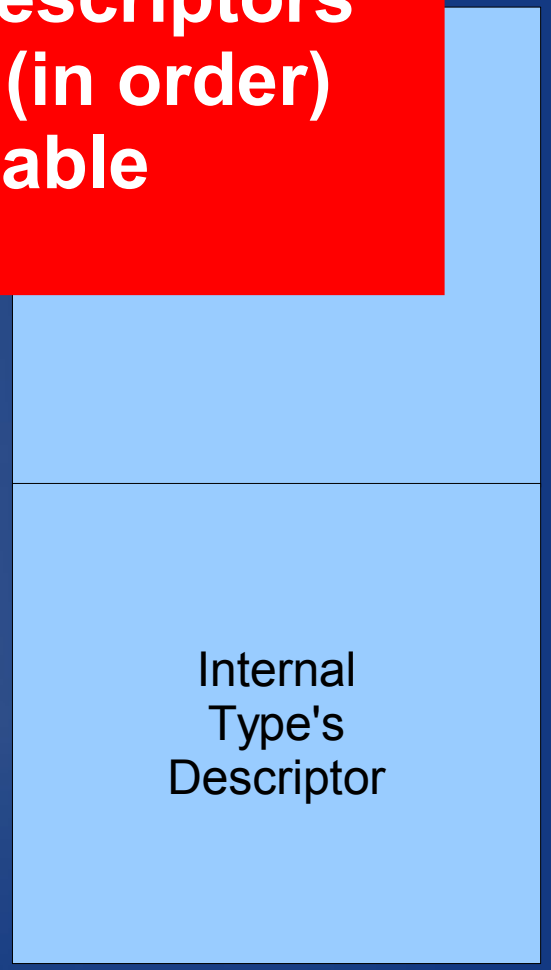
What is a type?

struct:{

integer:flags
@tonum
@getnum
@tostr
@tobool
@set
@get
@cast
@length *@size*
@exec
@setIndex *@getIndex* *@refIndex*
@setRange *@getRange* *@refRange*
@setMember *@getMember* *@refMember*
@newframe
@namedarg
@setflag
@parmoff
@exec

}

All type's descriptors are stored (in order) in a table



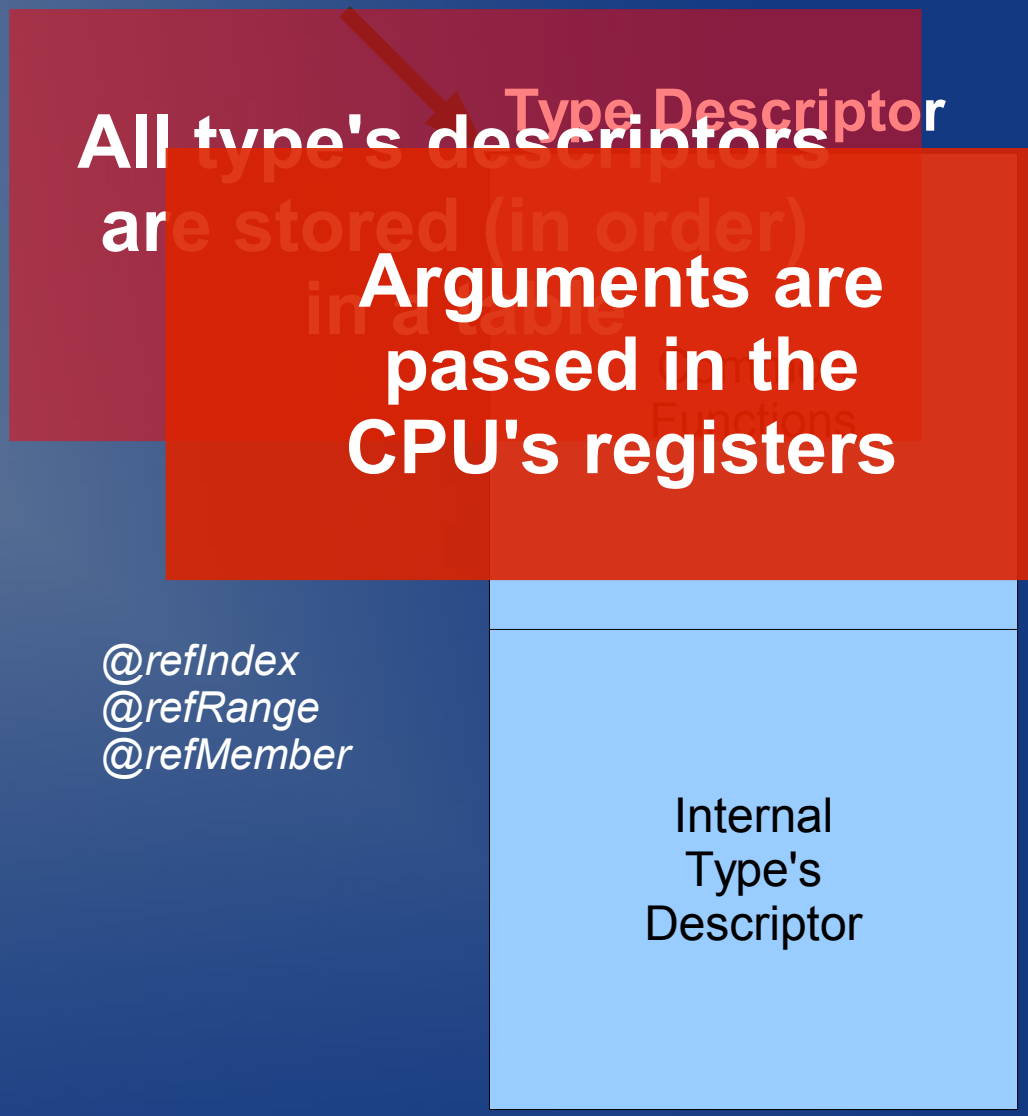
The Type system

What is a type?

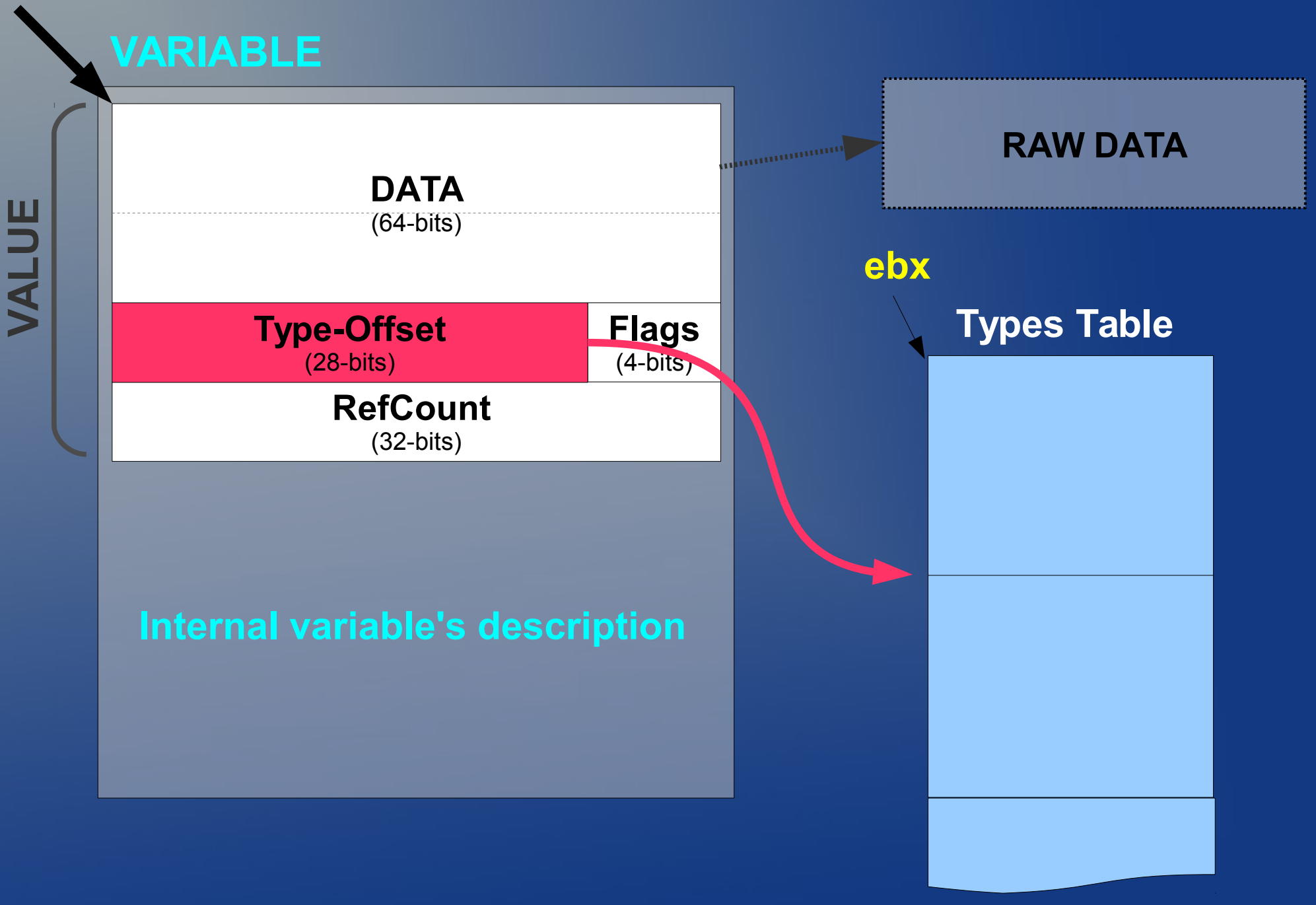
struct:{

integer:flags
@tonum
@getnum
@tostr
@tobool
@set
@get
@cast
@length *@size*
@exec
@setIndex *@getIndex* *@refIndex*
@setRange *@getRange* *@refRange*
@setMember *@getMember* *@refMember*
@newframe
@namedarg
@setflag
@parmoff
@exec

}

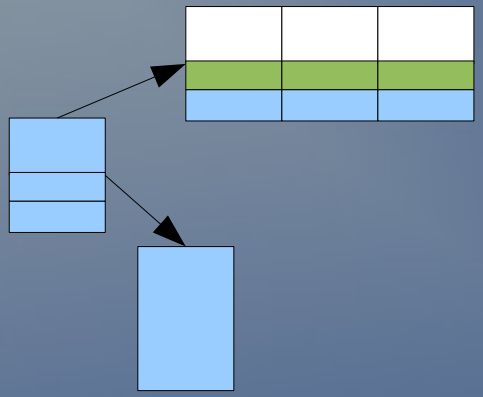


Values in memory



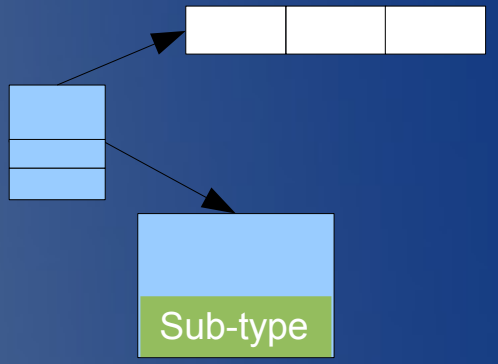
RAW Data storage

Undefined sub-type



type a array {3}
type b list {3}
type c hash
type c dict

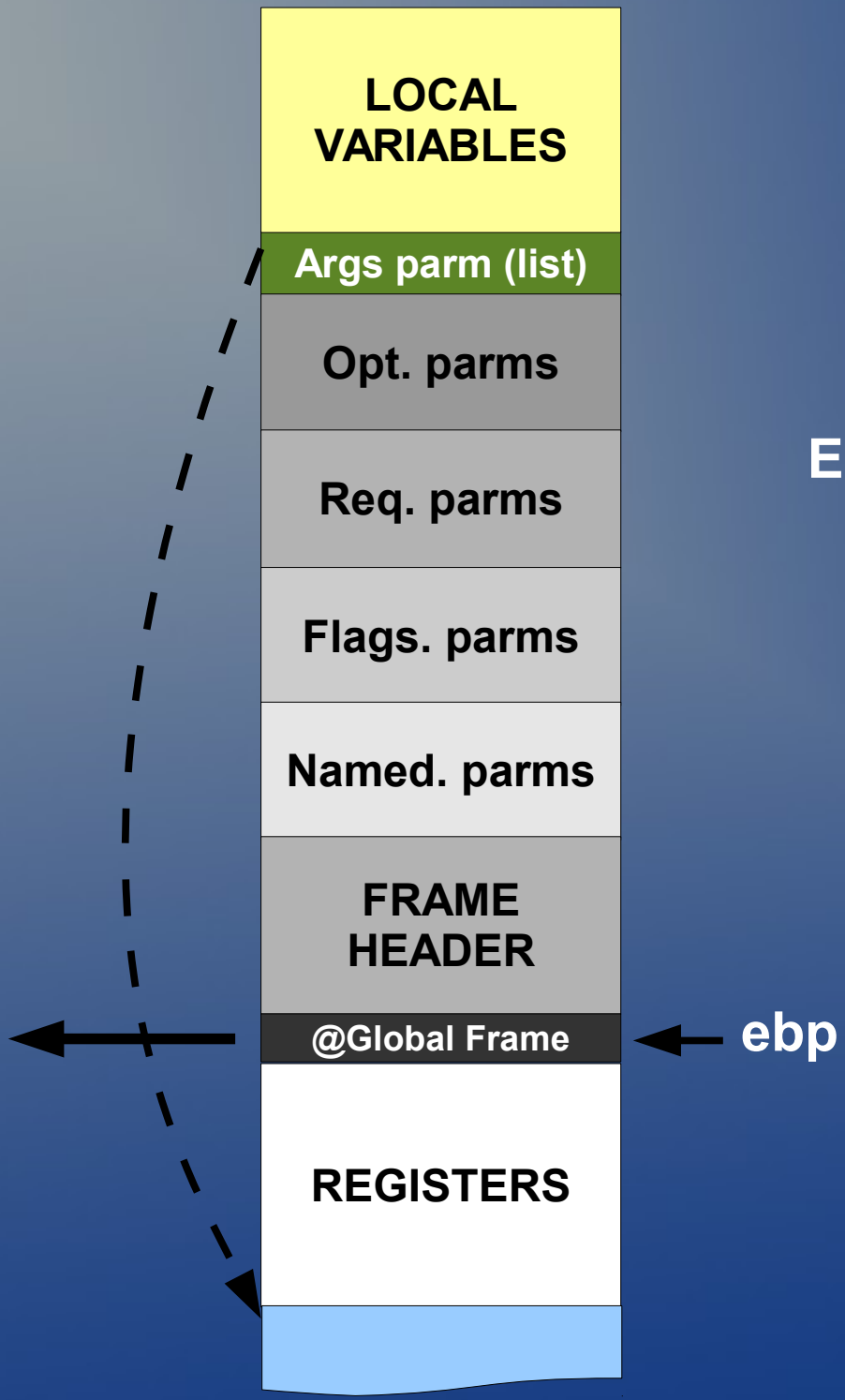
Defined sub-type



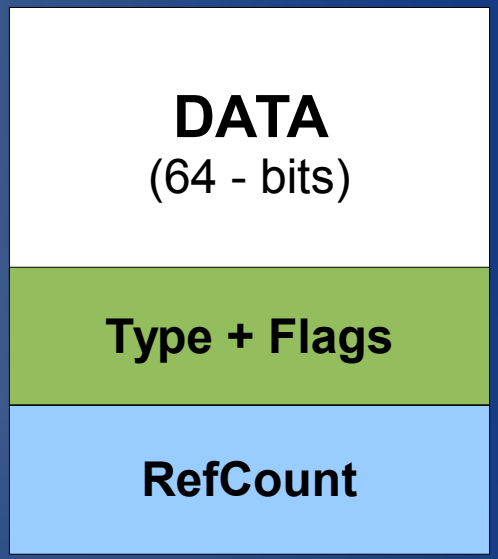
type a array {3 i32}
type b list {3 s}
type c hash {string}
type c dict {i8}

```
type s struct {
    i32:x
    f64:y
}
```

Execution frames



Each parm / register / local_variable:



Status

- Working “prototype”
- Initial performance tests show increases in performance between 2x and 25x vs. Tcl 8.6
- On-the-fly interpreter
- The posibility to run TyCL code in compiling time...this allows the modification of the whole compiler right at the spot.
- Experimenting with javascript as a target

To anyone interested … in any way :)

aabuss@otlettech.com
aabuss@gmail.com