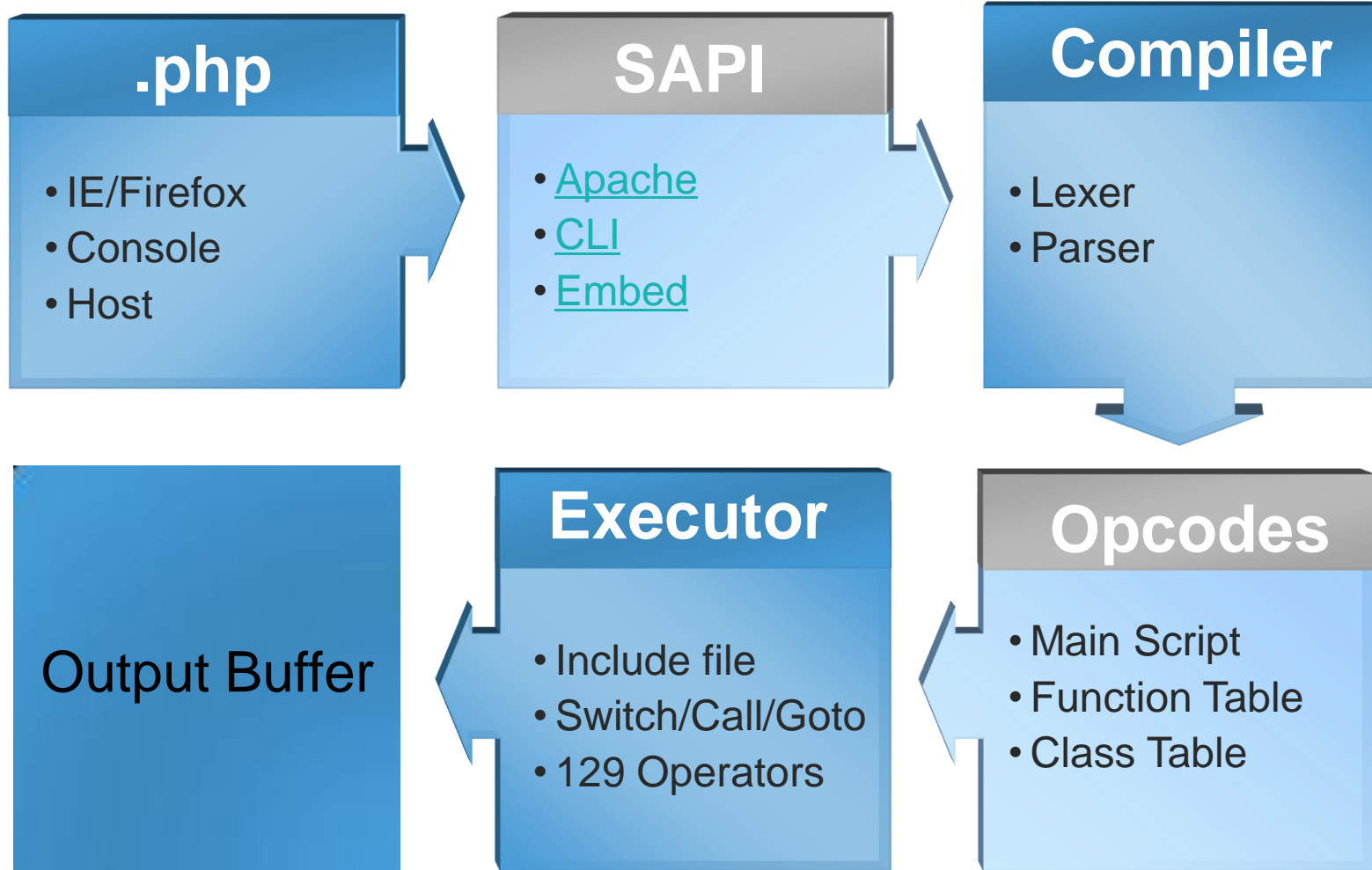
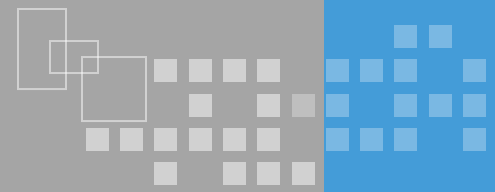


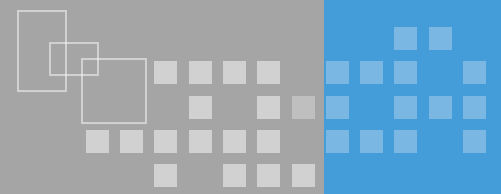


# PHP 运行机制初探

Ben ben.yan@msn.com

# Introduction





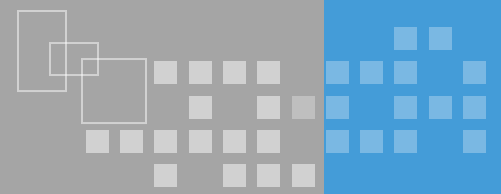
## ❖ **Mime type handler**

- AddType application/x-httpd-php .php
- AddType application/x-httpd-php-source .phps

## ❖ **Server context**

- Override php.ini (php\_value, php\_flag, etc)
- Environment variables(PHP\_SELF, etc)

## ❖ **Create Child Process/Thread**



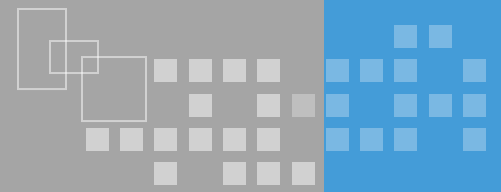
## ❖ **Mime type handler**

- AddType application/x-httpd-php .php
- AddType application/x-httpd-php-source .phps

## ❖ **Server context**

- Override php.ini (php\_value, php\_flag, etc)
- Environment variables(PHP\_SELF, etc)

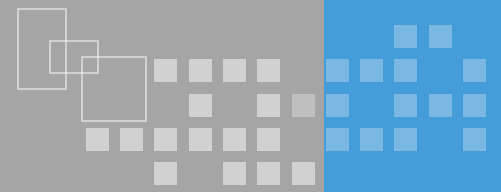
## ❖ **Create Child Process/Thread**



## ❖ CLI ≈ CGI SAPI

### ❖ differences

- start up in quiet mode by default
- plain text error message(no http header)
- `implicit_flush` always on
- `max_execution_time` is set to unlimited
- others



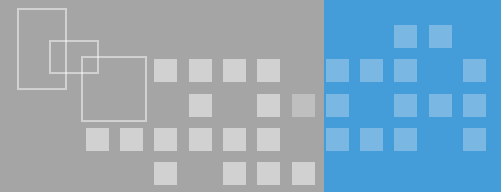
❖ **Embed = Mini CLI**

❖ **php5embed .lib**

❖ **example.c**

```
#include <php_embed.h>
int main (int argc, char *argv[]){
    PHP_EMBED_START_BLOCK(argc, argv)
    zend_eval_string("echo 'Hello World';", NULL, "Embedded Code" TSRMLS_CC);
    PHP_EMBED_END_BLOCK()
    return 0;
}
```

# Lexer(flex)



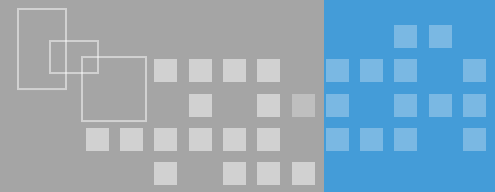
source: Zend/zend\_language\_scanner.l

```
<?php  
    $sum = 1 + 2;  
    echo '1+2='.$sum;  
?>
```



```
T_OPEN_TAG: '<?php '  
=  
T_LNUMBER: '1'  
+  
T_LNUMBER: '2'  
T_ECHO: 'echo"  
T_CONSTANT_ENCAPSED_STRING:  
    "1+2=".  
T_CLOSE_TAG: '?>'
```

# Parser(bison)



source: Zend/zend\_language\_parser.y

T\_OPEN\_TAG: '<?php '

=

T\_LNUMBER: '1'

+

T\_LNUMBER: '2'

T\_ECHO: 'echo'

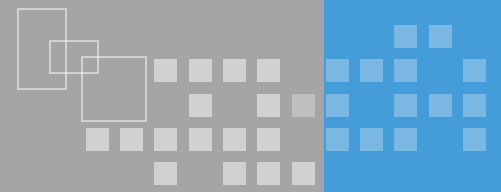
T\_CONSTANT\_ENCAPSED\_STRING:  
"1+2=".

T\_CLOSE\_TAG: '?>'



Opcode	Op1	Op2	Result
ADD	1	2	\$tmp0
ASSIGN	\$cv0(sum)	\$tmp0	\$var1
CONCAT	'1+2='	\$cv0(sum)	\$tmp2
ECHO	\$tmp2		
RETURN	1		

# Compiler



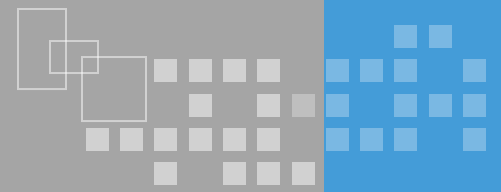
```
zend_op_array *(*zend_compile_file)(zend_file_handle *file_handle, int type TSRMLS_DC);
```

```
<?php  
    $sum = 1 + 2;  
    echo '1+2='.$sum;  
?>
```



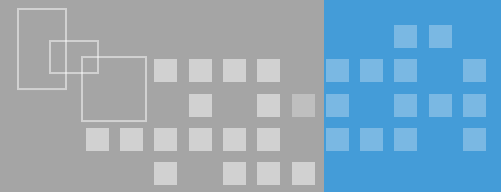
Opcode	Op1	Op2	Result
ADD	1	2	\$tmp0
ASSIGN	\$cv0(sum)	\$tmp0	\$var1
CONCAT	'1+2='	\$cv0(sum)	tmp2
ECHO	\$tmp2		
RETURN	1		

# Opcode



```
struct zend_op {  
    opcode_handler_t    handler;  
    znode result;  
    znode op1;  
    znode op2;  
    ulong extended_value;  
    uint lineno;  
    zend_uchar opcode;  
};
```

# Executor



```
void (*zend_execute)(zend_op_array *op_array TSRMLS_DC);
```

**SWITCH**

**CALL**

**GOTO**

PHP 4.x

```
switch (opcode){  
  case ZEND_ADD:  
    break;  
  case ZEND_CALL:  
    break;  
  ....  
}
```

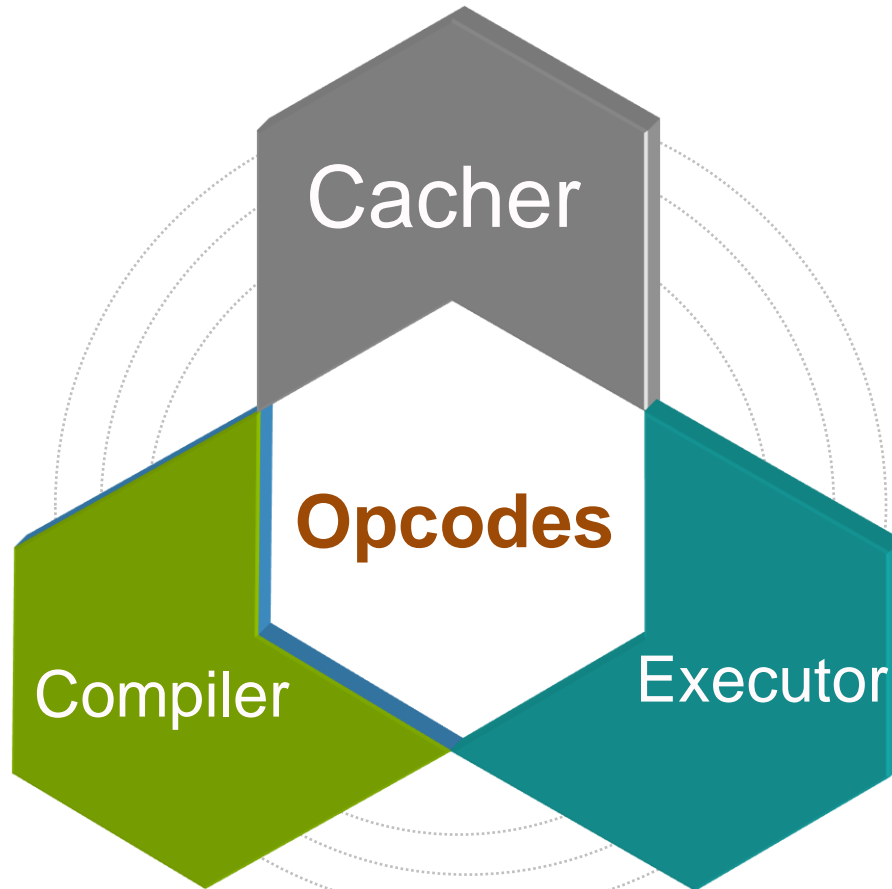
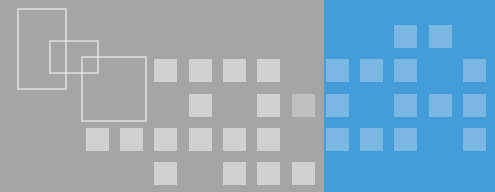
PHP 5.2

```
call ZEND_ADD_HANDLER()
```

PHP 5.2

```
switch (  
  case ZEND_ADD:  
    goto: zend_add  
)  
  
zend_add:  
//
```

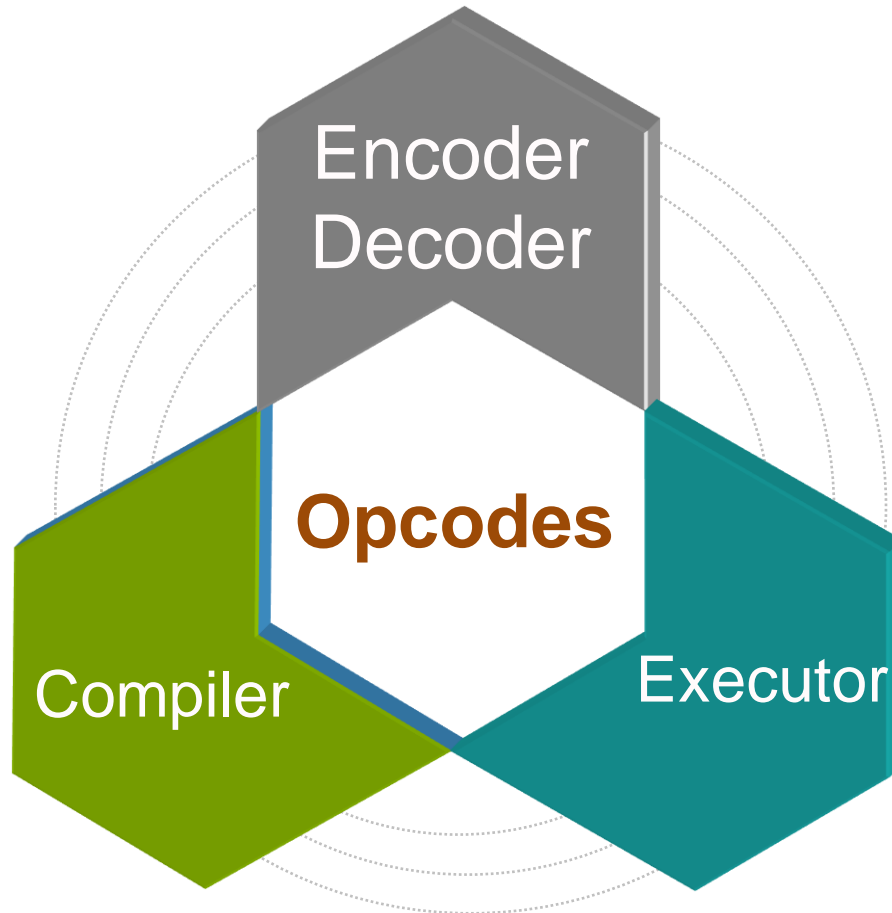
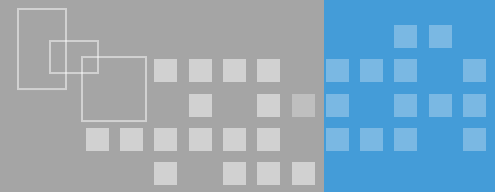
# Cacher



- APC
- eAccelerator
- XCache
- Zend Platform

```
PHP_MINIT_FUNCTION(xxx){  
    old_compile_file = zend_compile_file;  
    zend_compile_file = xxx_compile_file;  
}
```

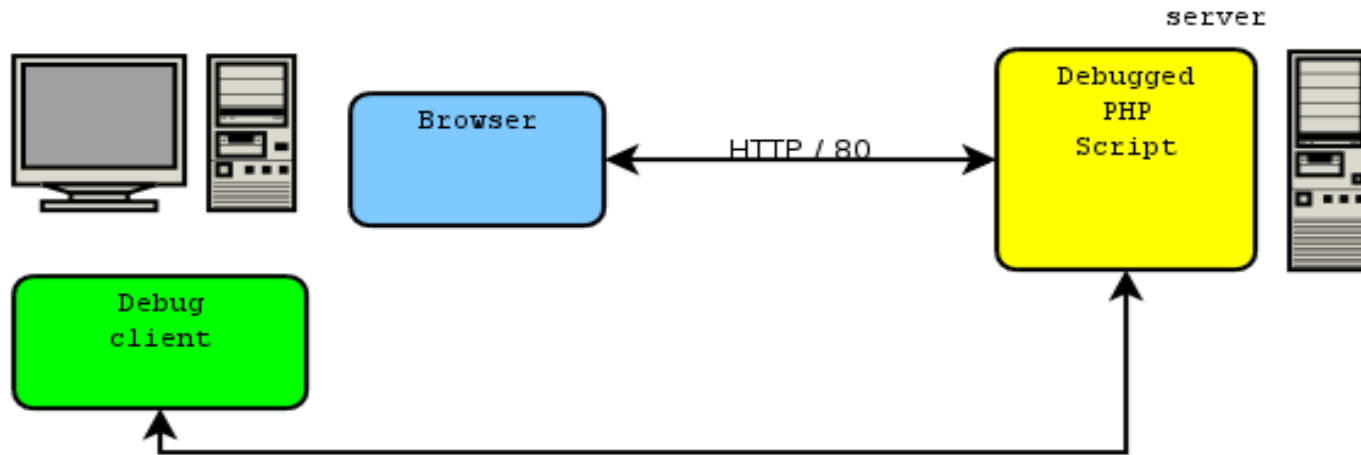
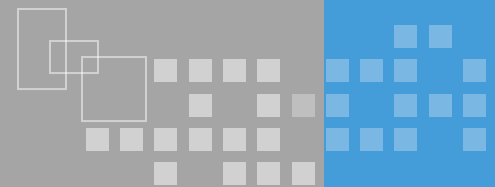
# Encoder / Decoder



- ZendGuard
- ionCube
- eAccelerator Encoder

NOT Encoder, BUT Obfuscator!

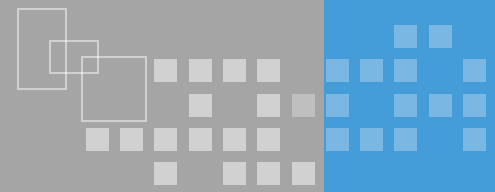
# Debugger



```
PHP_MINIT_FUNCTION(xxx){  
    old_execute = zend_execute;  
    zend_execute = xxx_execute;  
}
```

```
struct zend_extension {  
    message_handler_func_t message_handler;  
    op_array_handler_func_t op_array_handler;  
    statement_handler_func_t statement_handler;  
    fcall_begin_handler_func_t fcall_begin_handler;  
    fcall_end_handler_func_t fcall_end_handler;  
    op_array_ctor_func_t op_array_ctor;  
    op_array_dtor_func_t op_array_dtor;  
}
```

# Question





**Thank You !**