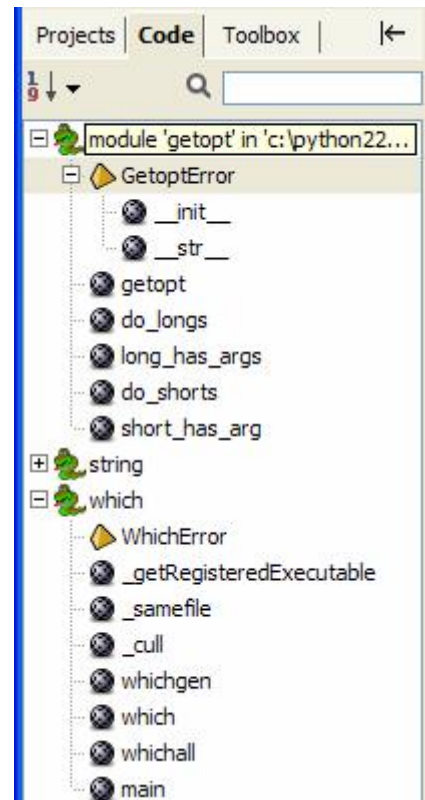




a division of SOPHOS

# Code Intelligence in Komodo



**Trent Mick**

**TrentM@ActiveState.com**

“Code Intelligence” is



a division of SOPHOS

§ **a Code Browser**

§ **autocomplete and call tips**

# What in it for me?



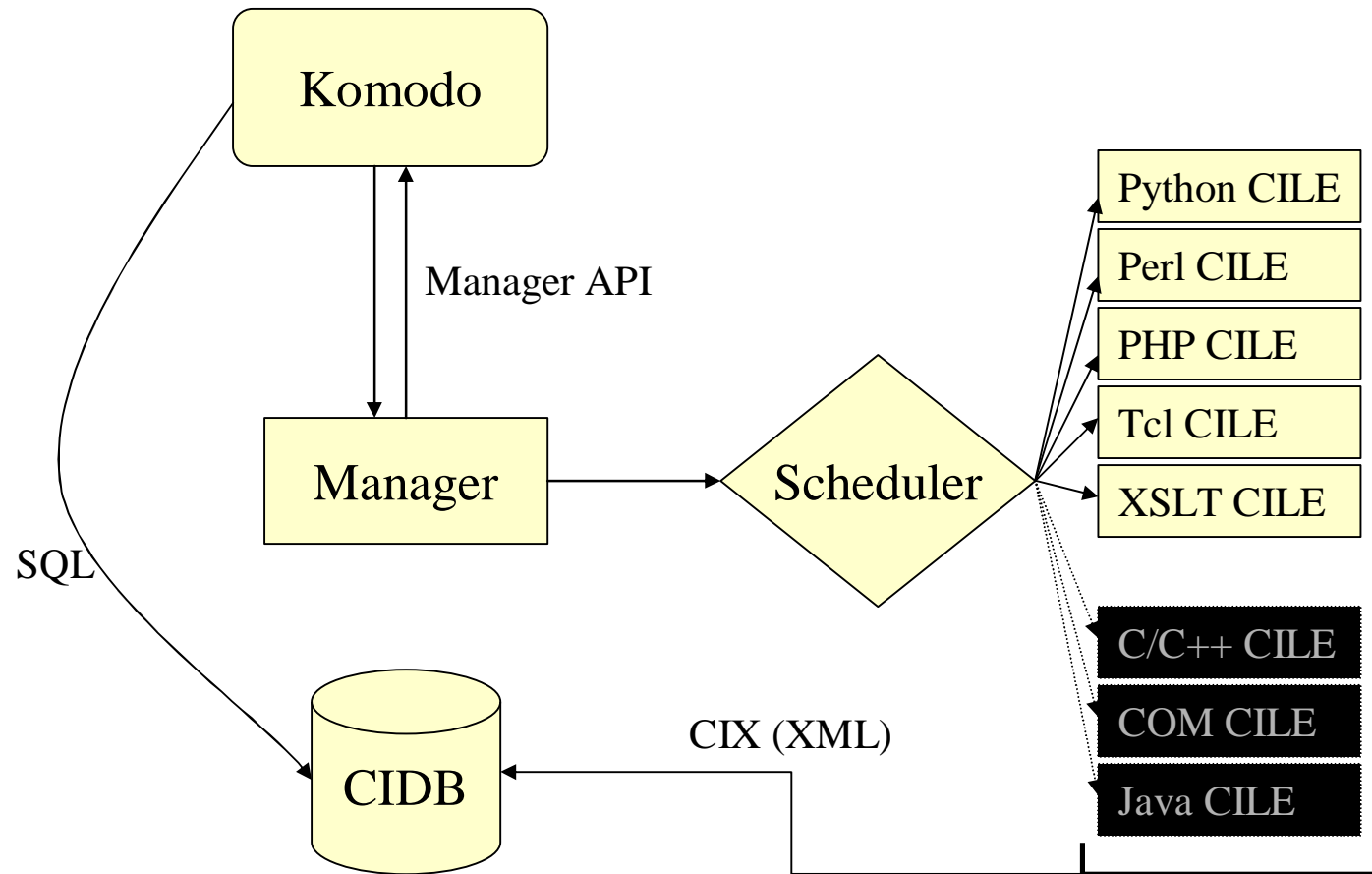
a division of SOPHOS

- § **See how an IDE Code Browser can be built**
- § **Learn about some useful Python packages**
- § **Couldn't find Mitch Kapor and lunch isn't served yet, so...**

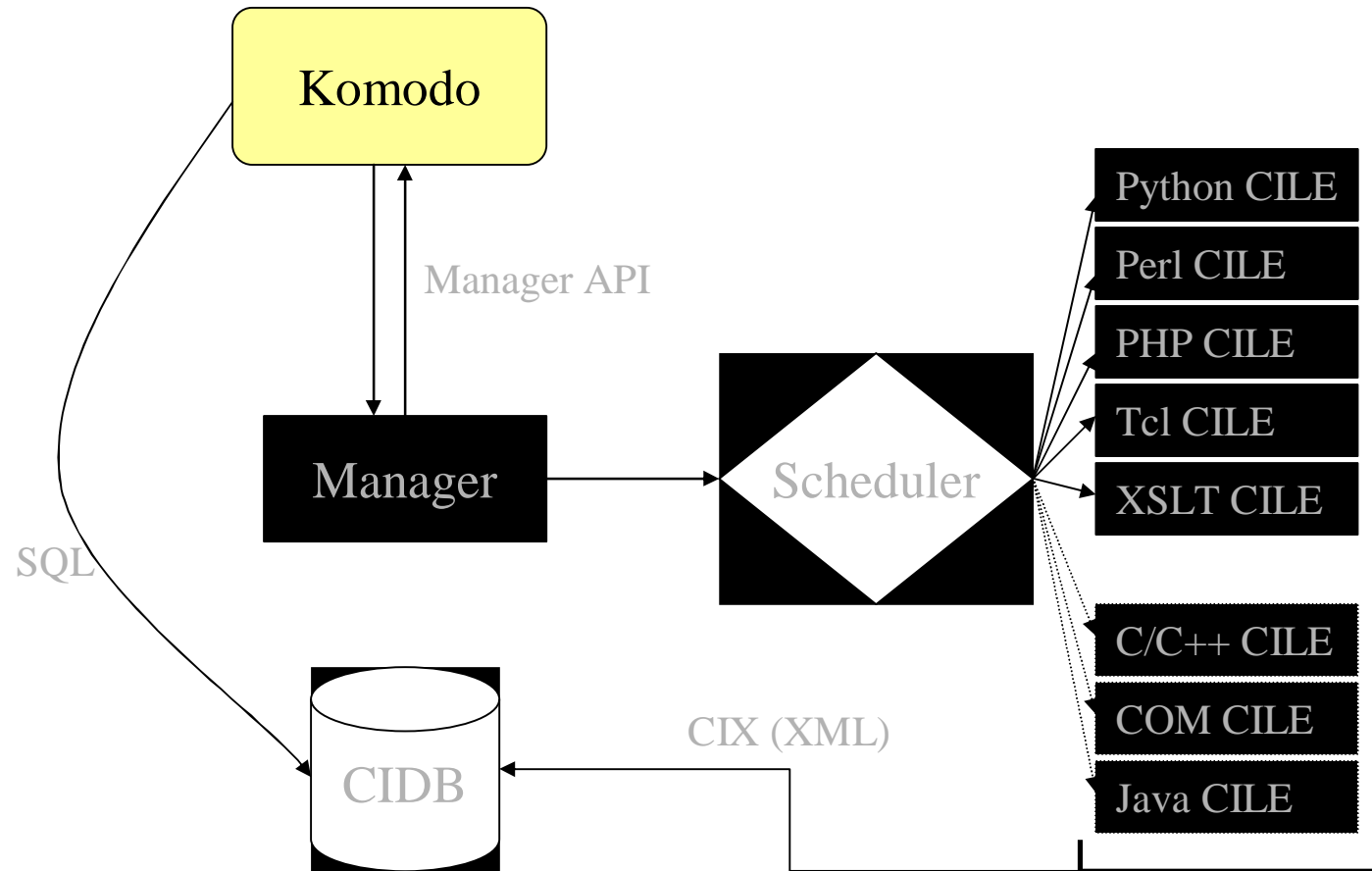
# Code Intelligence System Overview



a division of SOPHOS



# Application

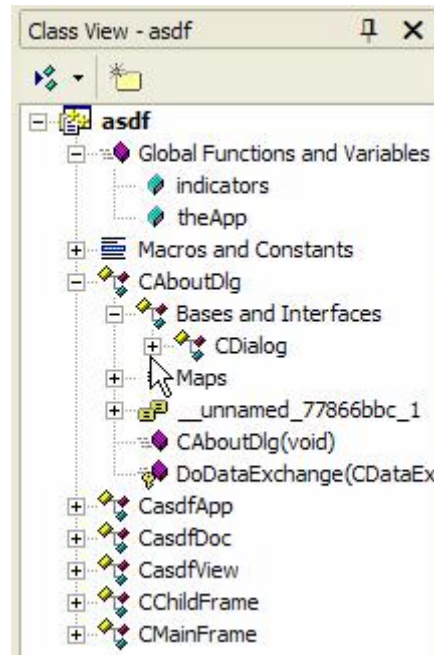


# What we're going for

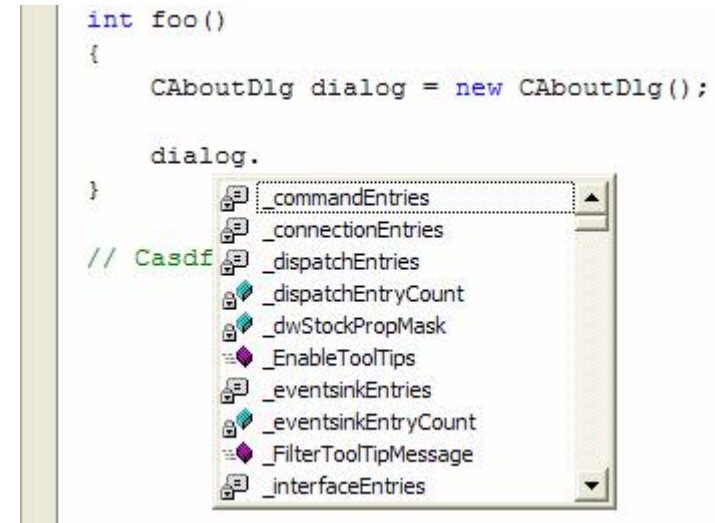


a division of SOPHOS

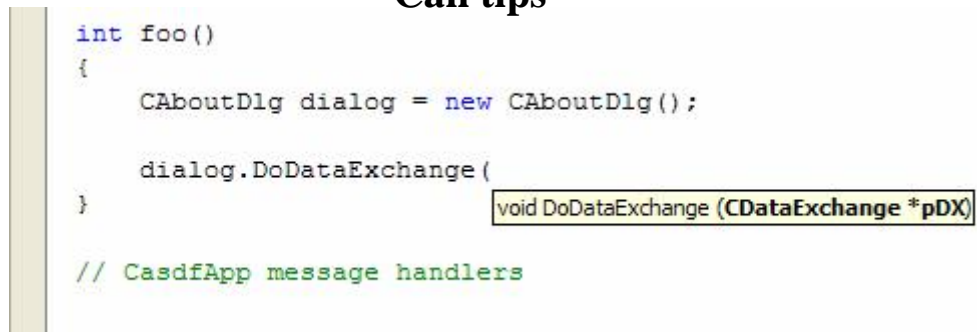
## Code Browser



## Autocomplete



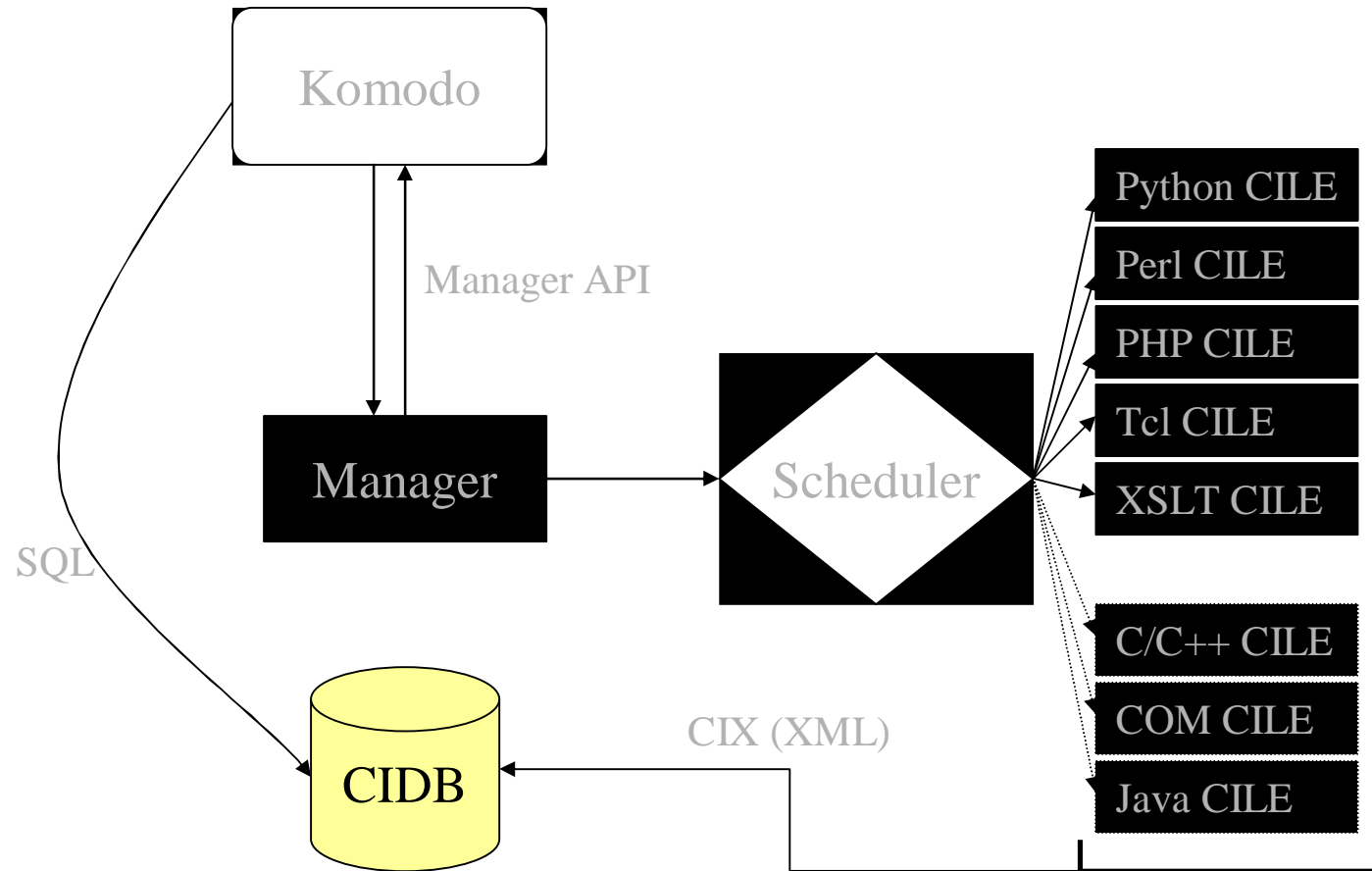
## Call tips



# Database



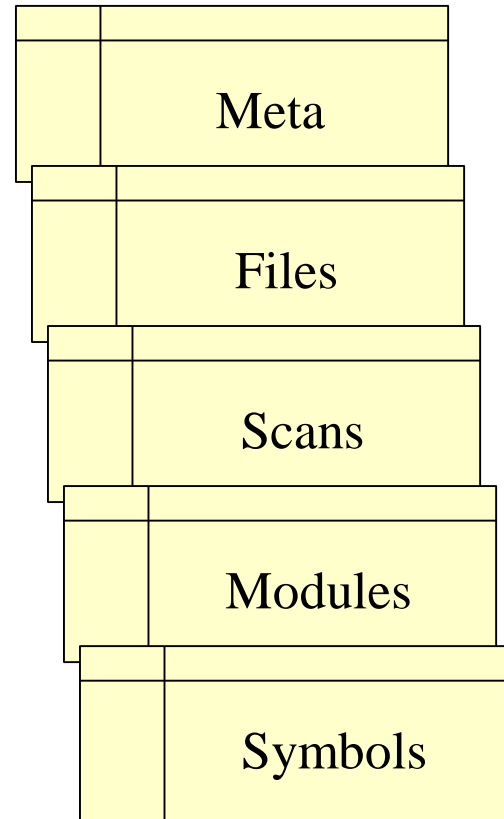
a division of SOPHOS



# Database



a division of SOPHOS



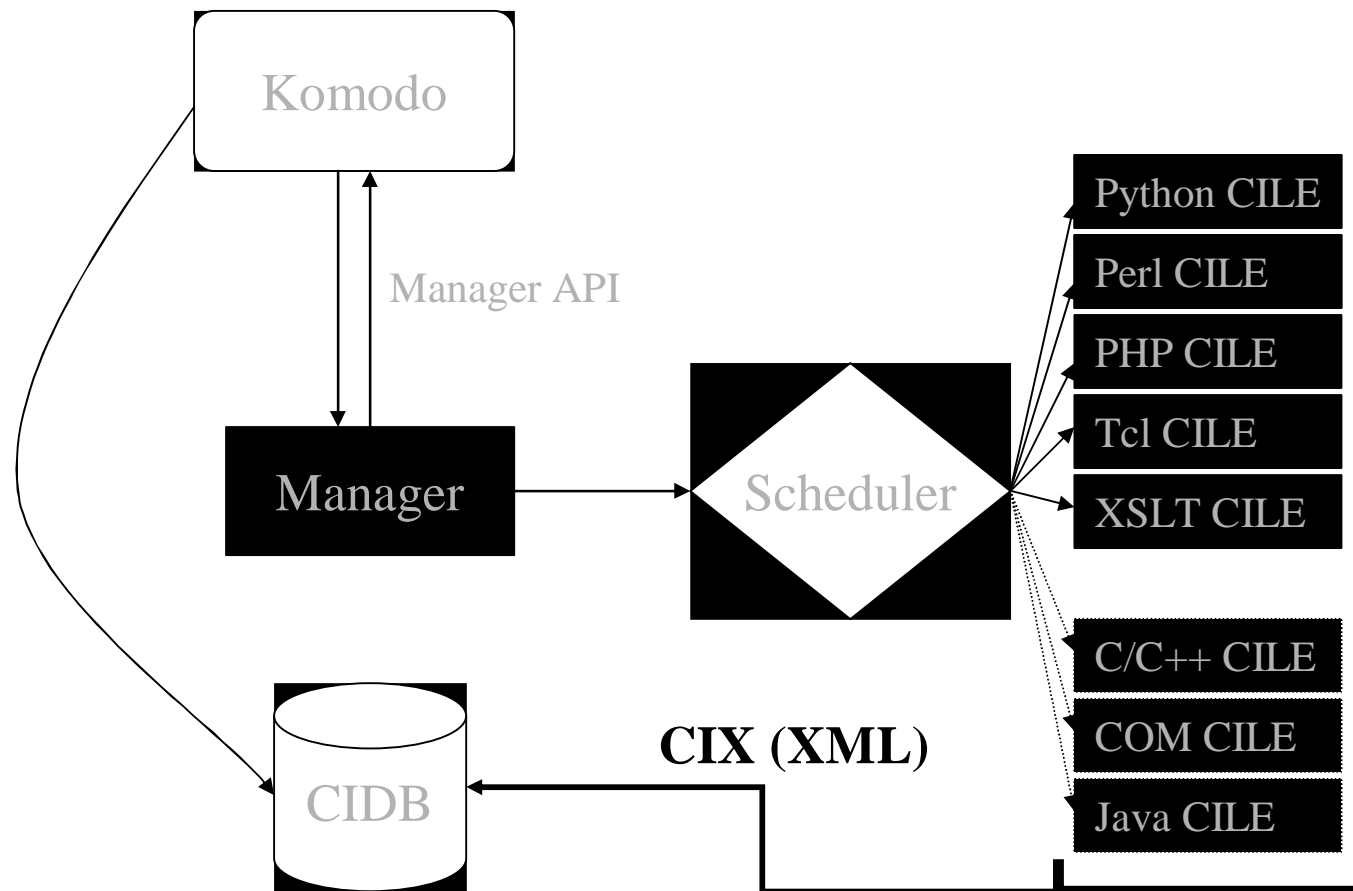
- § **SQLite for database**
- § **Separate “Files” and “Modules” for Perl and Tcl**
- § **Allow multiple scanners per file (e.g. Python, COM, and PyXPCOM)**
- § **Tip: include a “Meta” table**



# XML description format (CIX)



a division of SOPHOS



# XML description format (CI X)



a division of SOPHOS

```
<codeintel version="0.1">  
  <file path="foo.py" generator="Python"  
    language="Python"  
    md5="...">  
  ...  
</file>  
</codeintel>
```

- § **language-agnostic**
- § **storable and distributable**

# XML description format (CIX)



a division of SOPHOS

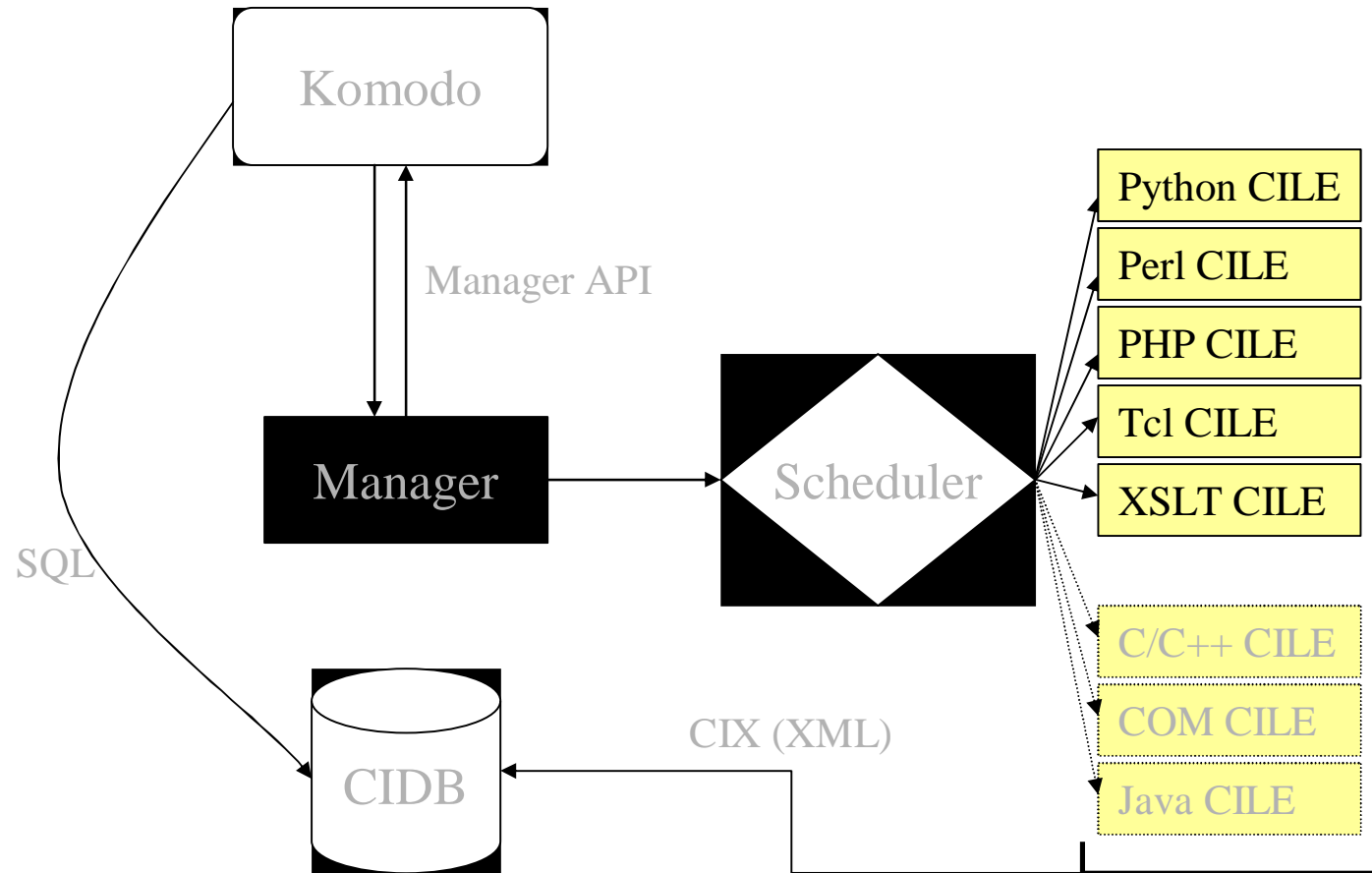
```
<codeintel version="0.1">
  <file path="string.py" ...>
    <module name="string">
      <function line="199" name="atof">
        <doc><![CDATA[atof(s) -> float ...]]></doc>
        <argument name="s"/>
      </function>
    </module>
  </file>
</codeintel>
```

§ **<module>, <function>, <class>, <interface>, <argument>, <doc>, <variable>, etc. tags define structure of the code**

# Language Engines (CILEs)



a division of SOPHOS



# Python Language Engine



a division of SOPHOS

## Python Source Code

```
198 # Convert string to float
199 def atof(s):
200     """atof(s) -> float
201
202     Return the floating point ...
203
204     """
205     ret
```

## AST from compiler

```
Function('atof', ['s'], [], 0, 'atof(s) -> float...',
  Stmt([
    Return(
      CallFunc(Name('_float'), [Name('s')], ...))])
```

## CIX

```
<module name="string">
  <function line="199" name="atof">
    <doc><![CDATA[atof(s) -> float ...]]></doc>
    <argument name="s"/>
  </function>
</module>
```



a division of SOPHOS

demo

# PHP Language Engine



a division of SOPHOS

## PHP Source Code

```
2 <?php
3 # atof(s) -> float
4 # ...
5 function atof($s) {
6     return _float($s);
7 }
8 ?>
```

parse with regular expressions

CIX

```
<module name="string.php">
  <function line="199" name="atof">
    <doc><![CDATA[atof(s) -> float ...]]></doc>
    <argument name="s"/>
  </function>
</module>
```



a division of SOPHOS

demo



# Perl, Tcl, XSLT Language Engines



a division of SOPHOS

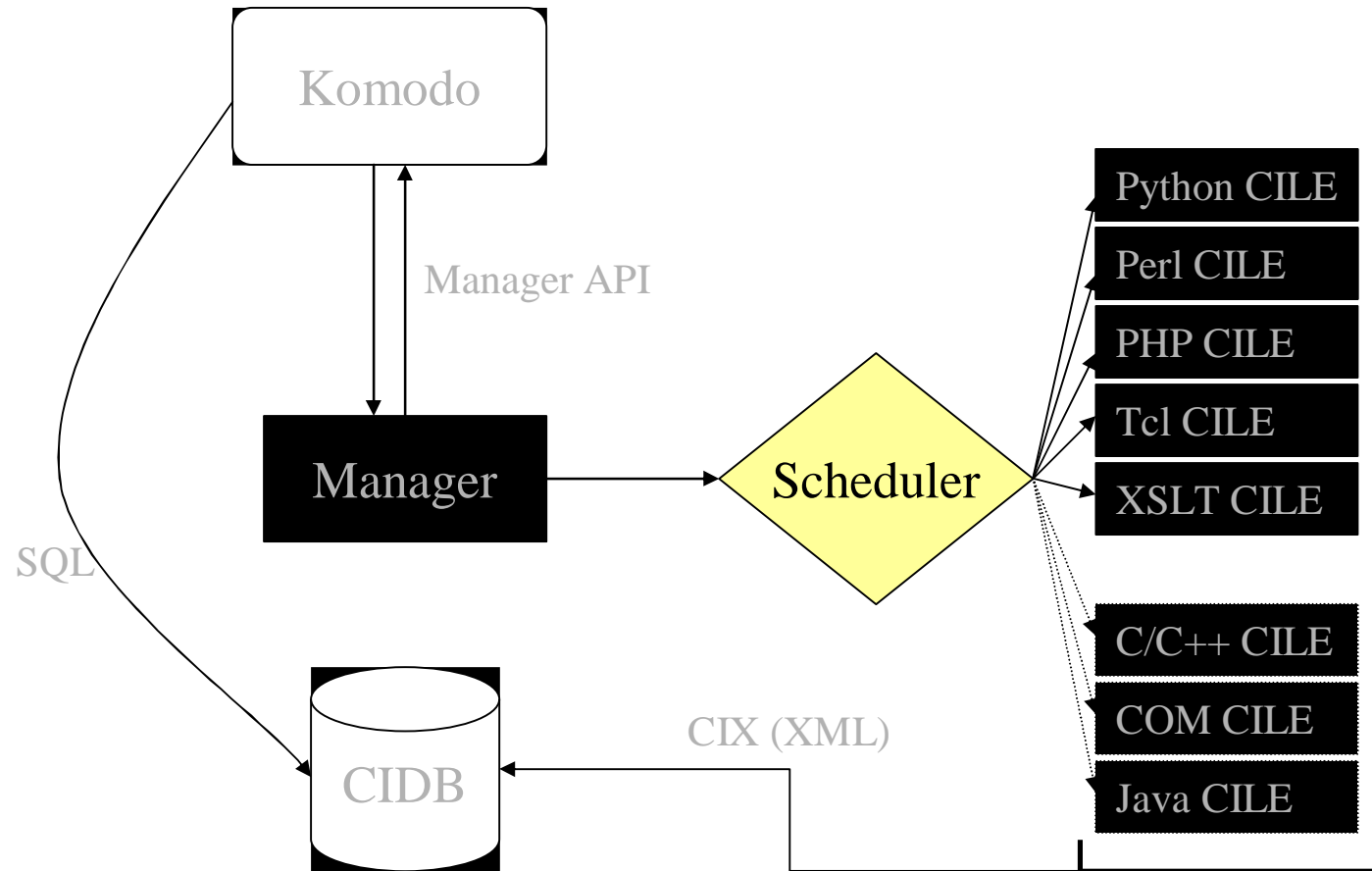
§ **The only thing that can parse Perl is Perl  
(but scintilla comes close)**

§ **Tcl can parse Tcl**

§ ***Tclwrap* can make an executable from Tcl**

§ ***libxml* can parse XSLT**

# Scheduler



# Scheduler



a division of SOPHOS

- § **Info must update dynamically as the user types, but not interfere with editing: i.e. asynchronous**
- § **“Scheduler” is a thread that manages a small pool of “Scan Runner” threads to scan file content.**



a division of SOPHOS

demo

Tip: import Queue



a division of SOPHOS

## § From the Python Library Reference:

The *Queue* module implements a multi-producer, multi-consumer FIFO queue. It is especially useful in threads programming when information must be exchanged safely between multiple threads.

### Manager

```
s = Scheduler()
s.queue.put("scan string.py")
s.queue.put("scan string.php")
```

### Scheduler

```
while 1:
    request = self.queue.get()
    # process request...
```

Tip: import Queue



a division of SOPHOS

## § This is a Priority Queue:

```
class PriorityQueue(Queue.Queue):  
    def _put(self, item):  
        bisect.insort(self.queue, item)
```

# Autocomplete/Calltips



a division of SOPHOS

## § **Type inferencing**

§ c.f. Mike Salib's "Starkiller" talk tomorrow

## § **Automatic triggering (language dependent):**

§ Python: *foo.*

§ Perl: *\$foo->*

§ PHP: *pri*

# Possibilities



a division of SOPHOS

## § **Other language engines:**

§ C/C++, Java, COM, XML/XHTML

## § **Distributing CIX files for binary packages**

## § **Custom user data in the database:**

§ method frequency tracking



Thank you



**Questions?**

**Trent Mick (TrentM@ActiveState.com)**