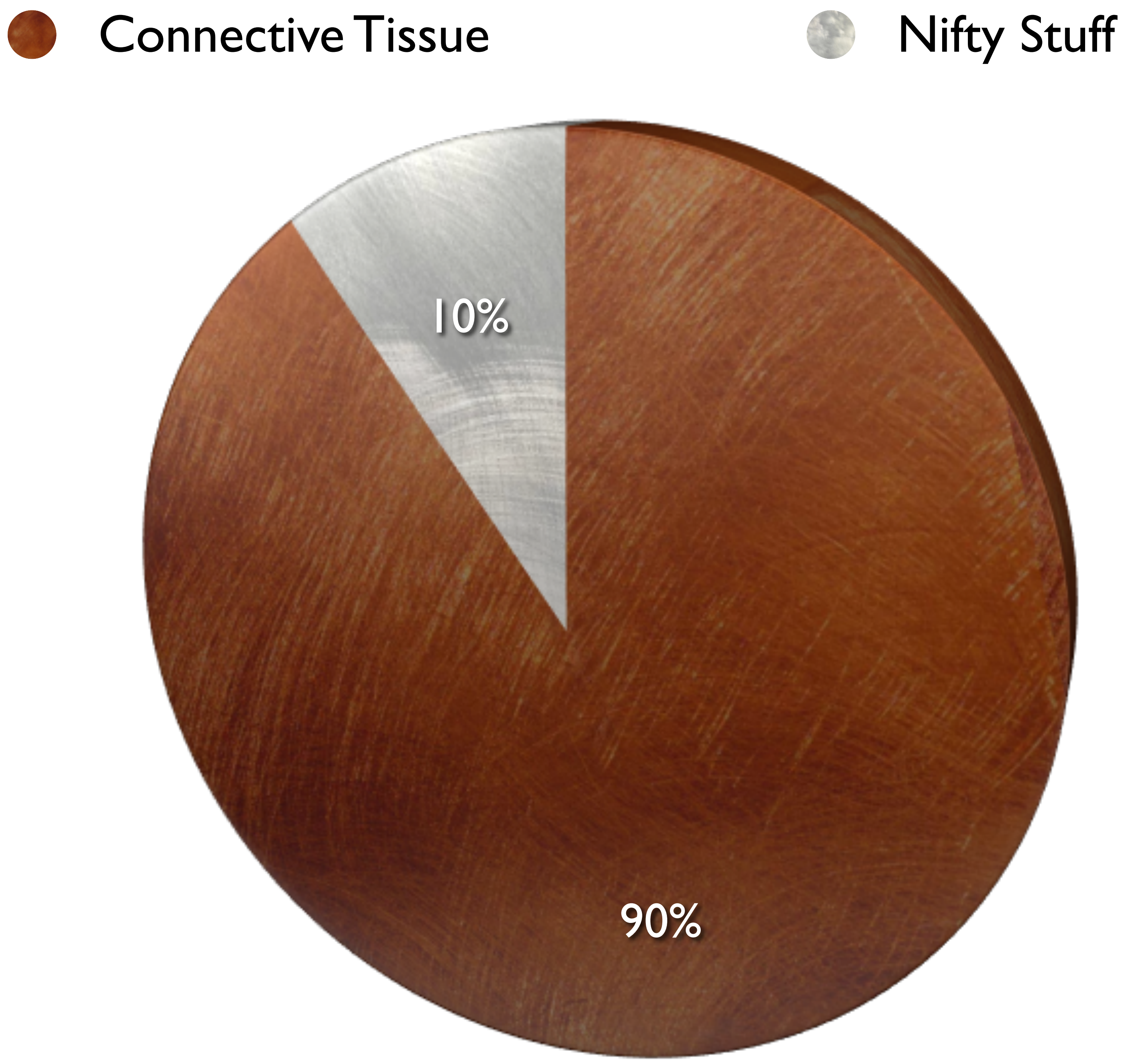
TAO/TK

TAO/TK

- Connective Tissue
- Nifty Stuff

TAO/TK



Features

Features

- Property Inheritance

Features

- Property Inheritance
- Option Handling

Features

- Property Inheritance
- Option Handling
- Notification Handling

Features

- Property Inheritance
- Option Handling
- Notification Handling
- Megawidget Toolkit

Features

- Property Inheritance
- Option Handling
- Notification Handling
- Megawidget Toolkit
- Sqlite Backend

Features

- Property Inheritance
- Option Handling
- Notification Handling
- Megawidget Toolkit
- Sqlite Backend
- Method Ensembles

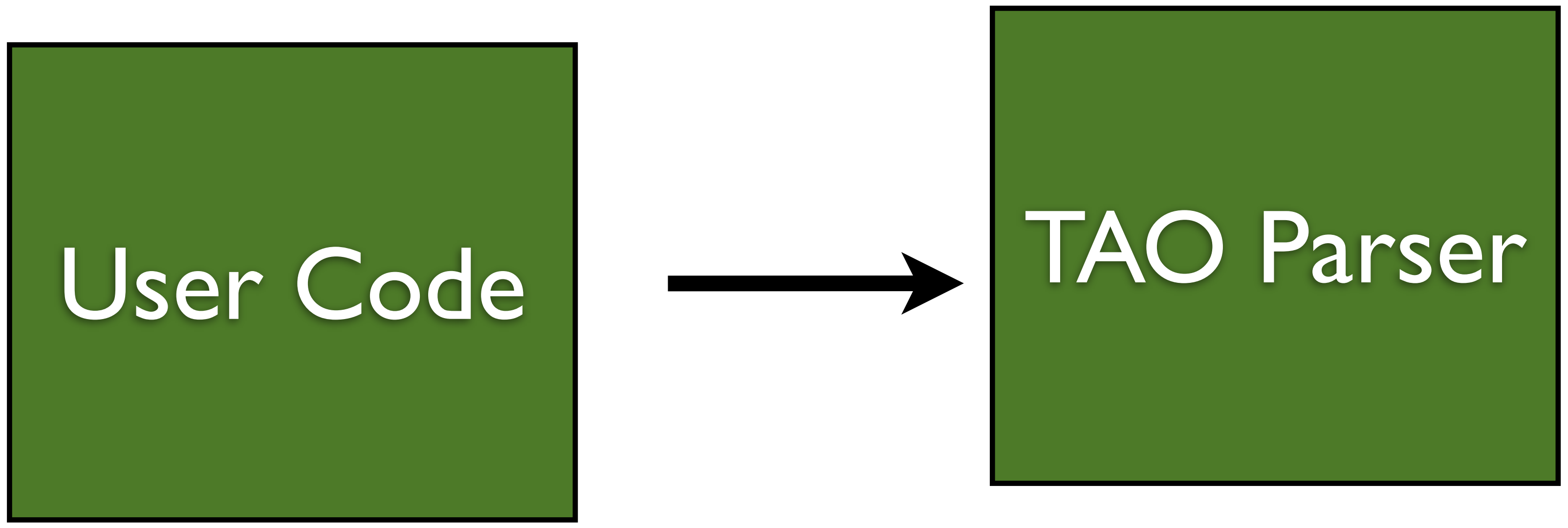
Features

- Property Inheritance
- Option Handling
- Notification Handling
- Megawidget Toolkit
- Sqlite Backend
- Method Ensembles
- Class Instance Method Inheritence

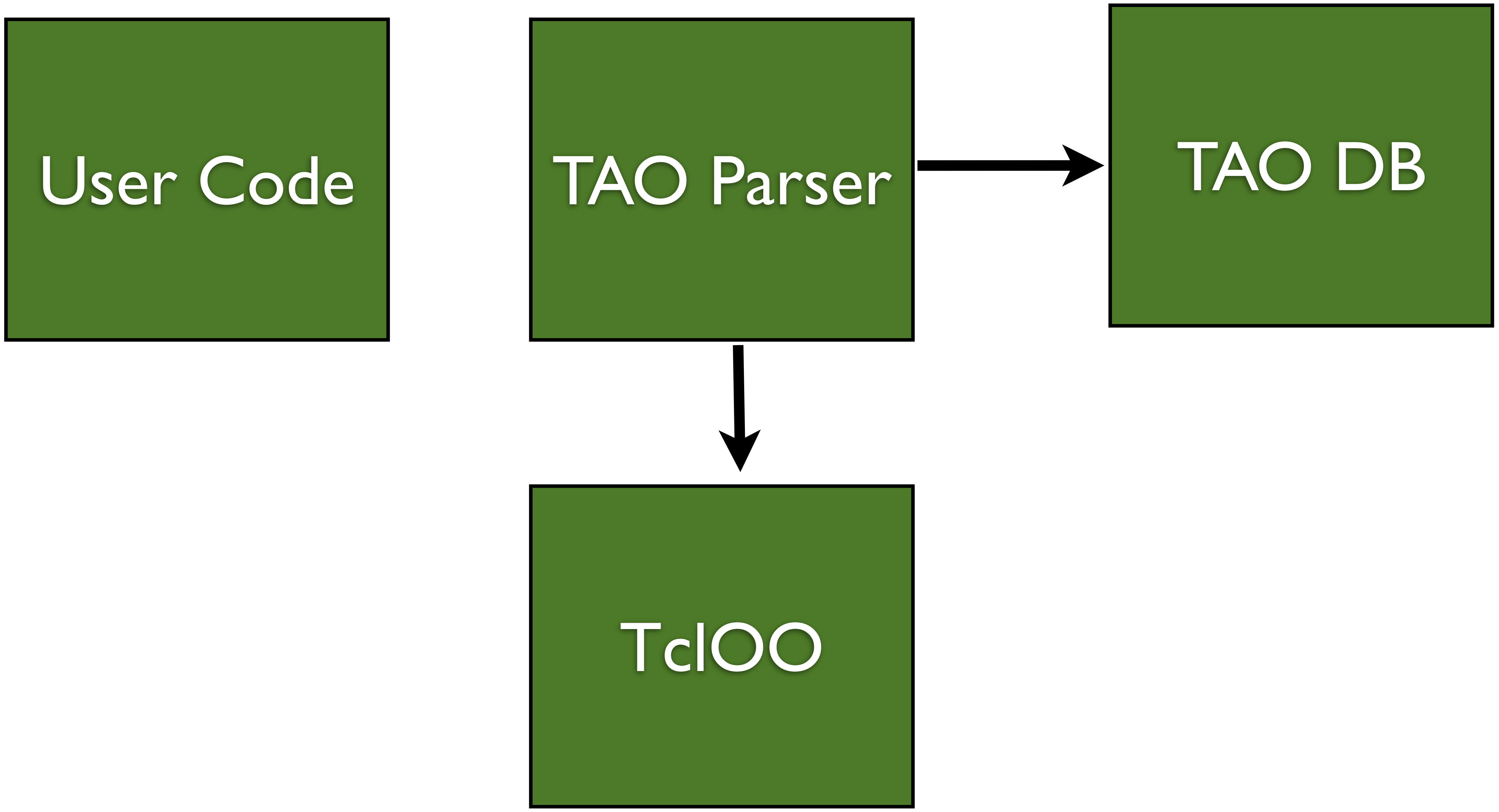
TAO/TK

User Code
TAO Parser

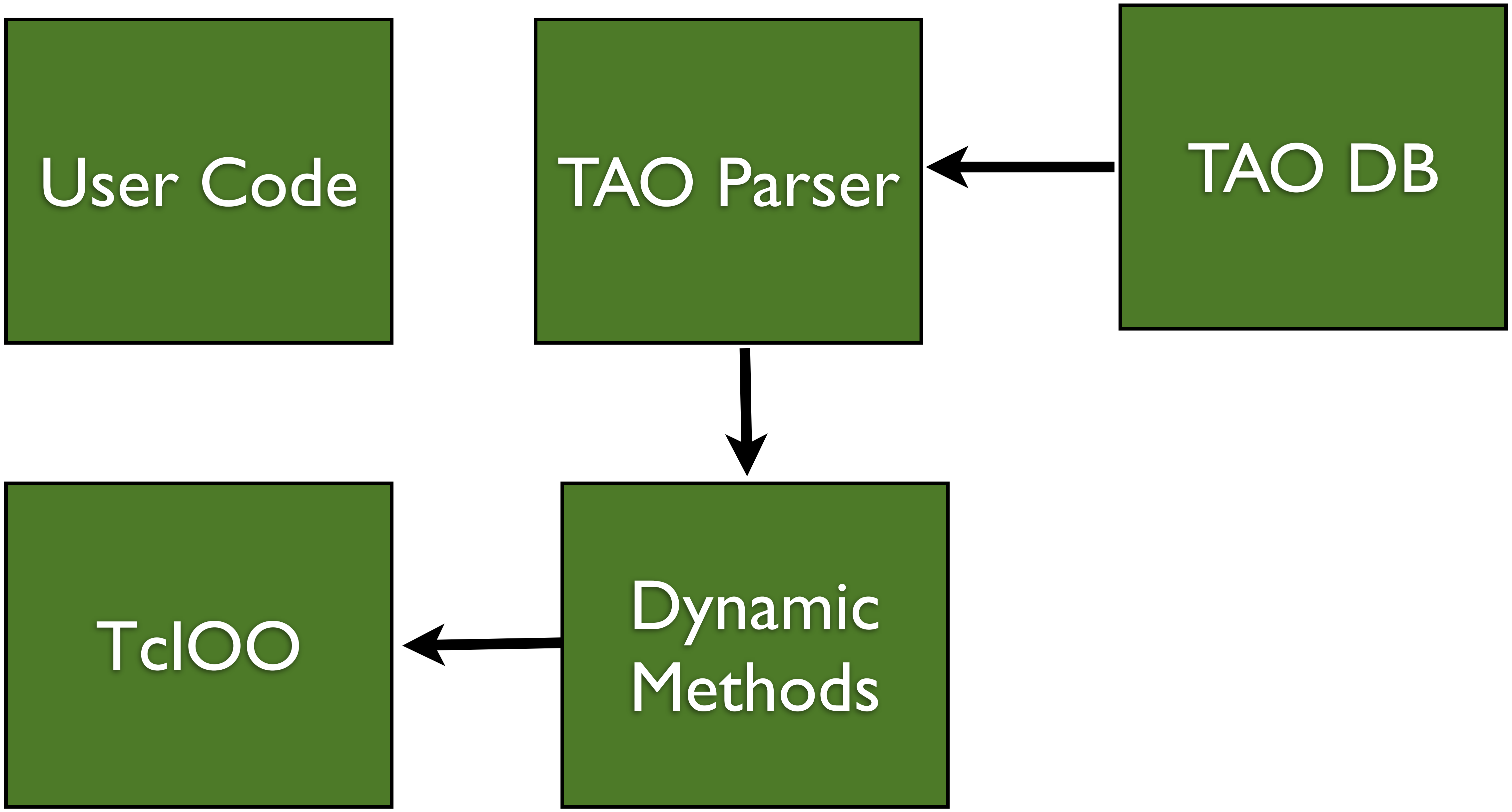
TAO/TK



TAO/TK



TAO/TK



TAO/TK

- All TAO/TK code ultimately becomes pure TclOO code

Notation

- TAO/TK Tries very hard to mimic TclOO syntax

```
tao::class foo {
}
```

::tao::class is an amalgamation of *oo::class create* and *oo::define*

```
proc ::tao::class {name definition} {
...
}
```

The preparser is a proc in the ::tao namespace

```
tao::class foo {
  superclass bar
}
```

In most respects it defers back to TclOO

```
tao::class foo {
  superclass bar

  method ipso {} {
    return lorem
  }
}
```

In most respects it defers back to TclOO

```
tao::class foo {
  superclass bar

  property color green
}
```

But it adds additional keywords

TAO/TK Keywords

- class_method
- option
- property
- signal
- variable

Keyword: property

- A chunk of data that is passed to children
- Assumed to be a constant string, but it can be other things
- Tracked in sqlite
- Accessed from the “property” dynamic method

Keyword: option

- Actually a special type of property
- Defined as a key/value description of option properties
- Options are instantiated with default values inside the constructor
- Accessed from the configure and cget methods

Keyword: signal

- Yet another “property” in disguise
- Also a key/value description
- Define chunks of code that will ultimately become the objects signal pipeline

Keyword: class_method

- Methods that are given to the class object itself
- Unlike calling oo::objdefine, **class_method** is passed from ancestor to descendent

Keyword: variable

- (Yes yes... I am breaking compatibility with conventional TclOO)
- Declares a variable and it's default value to be instantiated in the constructor
- Still **another** property in disguise
- (I'm taking suggestions for a better term.)

Method Ensembles

- TAO/TK has a rudimentary system for nesting methods
- Submethods can take arguments

```
tao::class foo {
  method hello::world {} {
    puts “Hello World”
  }
}
```

Methods with a :: in the name are interpreted as method ensembles

```
tao::class foo {
  method hello::kitty {} {
    puts “Hello Kitty”
  }
}
```

```
info class definition ::foo hello
{method args} {
  switch $method {
    <list> {
       return {dolly kitty world}
    }
    dolly { ... }
    kitty { ... }
    world { ... }
    default { ... }
  }
}
```

The TAO parser implements ensembles as a switch.

Mother of all Classes

- All TAO objects inherit class moac, the Mother of all Classes
- moac provides a suite of functions and reference implementations

Signal Pipeline

- Signals are snippets of code that are executed in response to events, but after an idle event has occured
- Signals can trigger other signals, but for each pipeline execution each signal's snippet will only run once

Example

```
tao::class fisherman {

  signal fish {
  }

  signal cut_bait {
  }
}
```

- The fisherman has a 2 stage pipeline:
  - Fish
  - Cut bait

Example

```
tao::class fisherman {

 signal fish {
   follows cut_bait
   action {my fish}
 }

 signal cut_bait {
  action {my cut_bait}
 }
}
```

- Cut bait must happen before fish

Example

```
tao::class fisherman {

 signal fish {
   triggers cut_bait
   follows cut_bait
   action {my fish}
 }

 signal cut_bait {
  action {my cut_bait}
 }
}
```

- All calls to fish should trigger a cut bait

```
fisherman create gordan
gordan signal fish
> gordan is cutting bait
> gordan is fishing
```

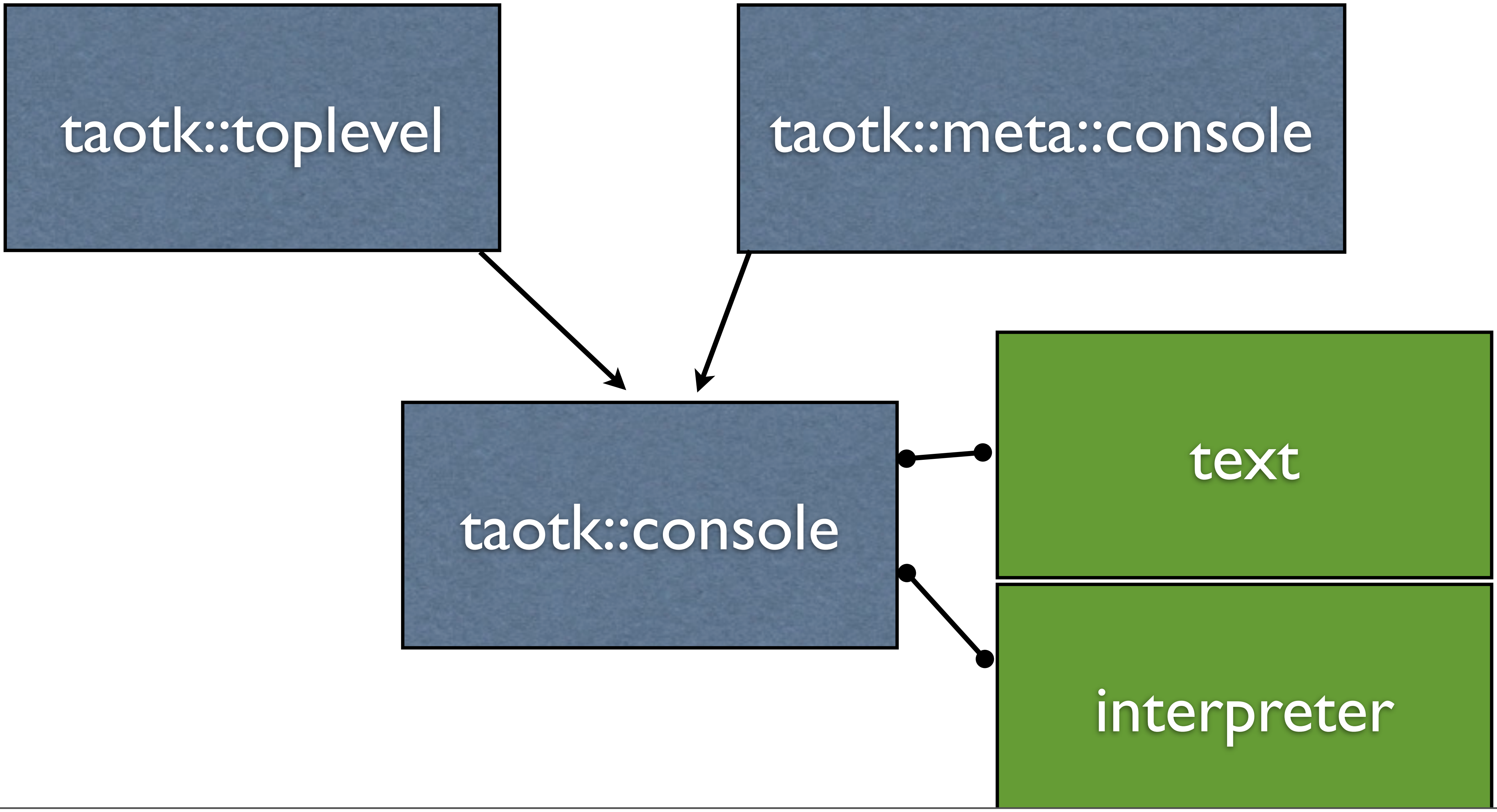
```
gordan signal fish
gordan signal fish
gordan signal fish
gordan signal fish
gordan signal fish
gordan signal fish
gordan signal fish

> gordan is cutting bait
> gordan is fishing
```

Megawidgets

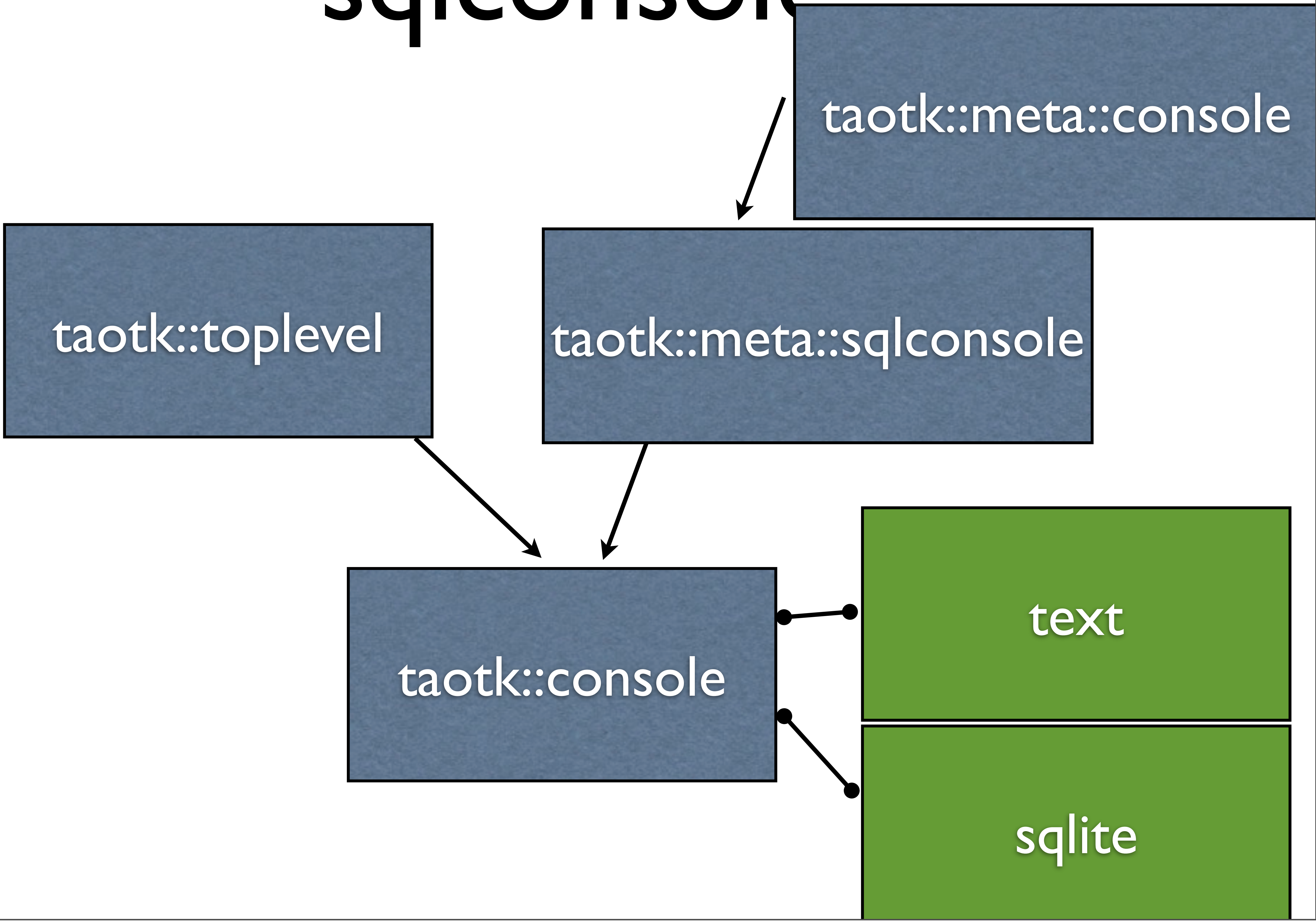
- Meta Classes
- Classes Ready to Use
- Dynamic GUI Magic

console


taotk::toplevel
taotk::meta::console
taotk::console
text
interpreter

console (Demo)

sqlconsole


taotk::meta::console
taotk::toplevel
taotk::meta::sqlconsole
taotk::console
text
sqlite

sqlconsole (Demo)

Dynamic Gui Magic

- TAO/TK Ships with a suite if ready to use data entry elements
- They must conform to the template:
  - Each UI element is a field in an array
  - Each field has a key/value description

Dynamic Gui Magic (Demo)