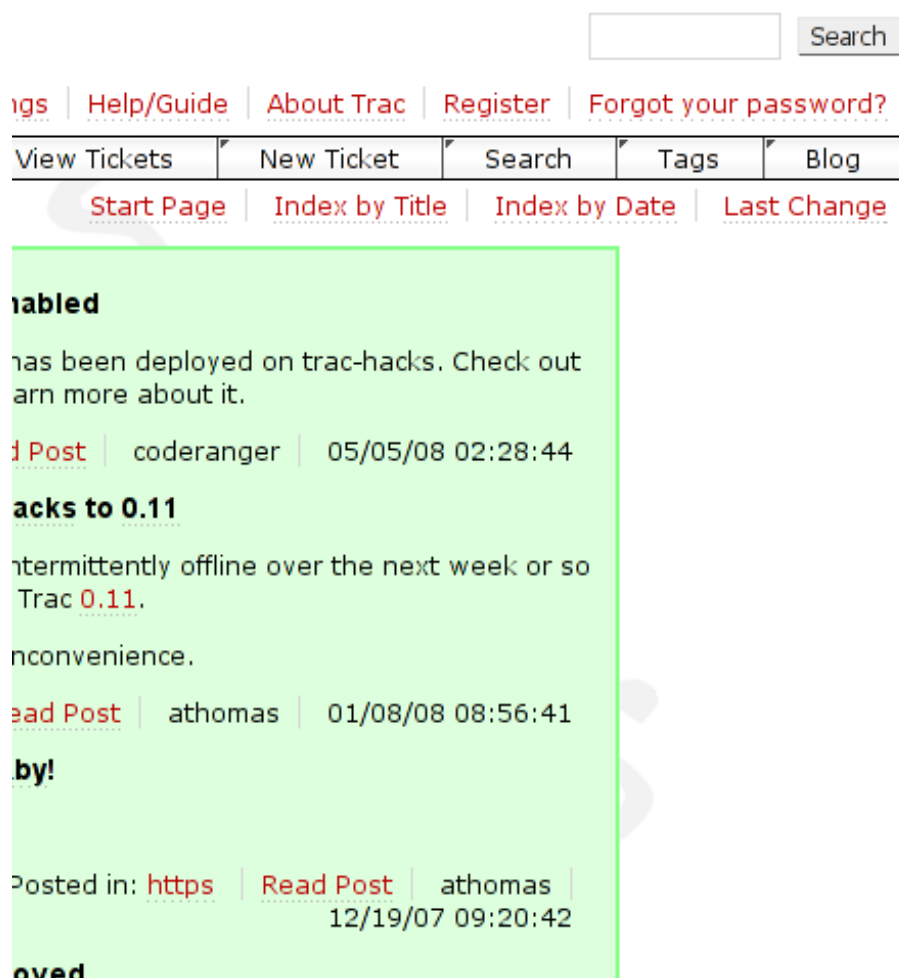


# Плагины и интерфейсы в Python

Михаил Гусаров <dottedmag@dottedmag.net>

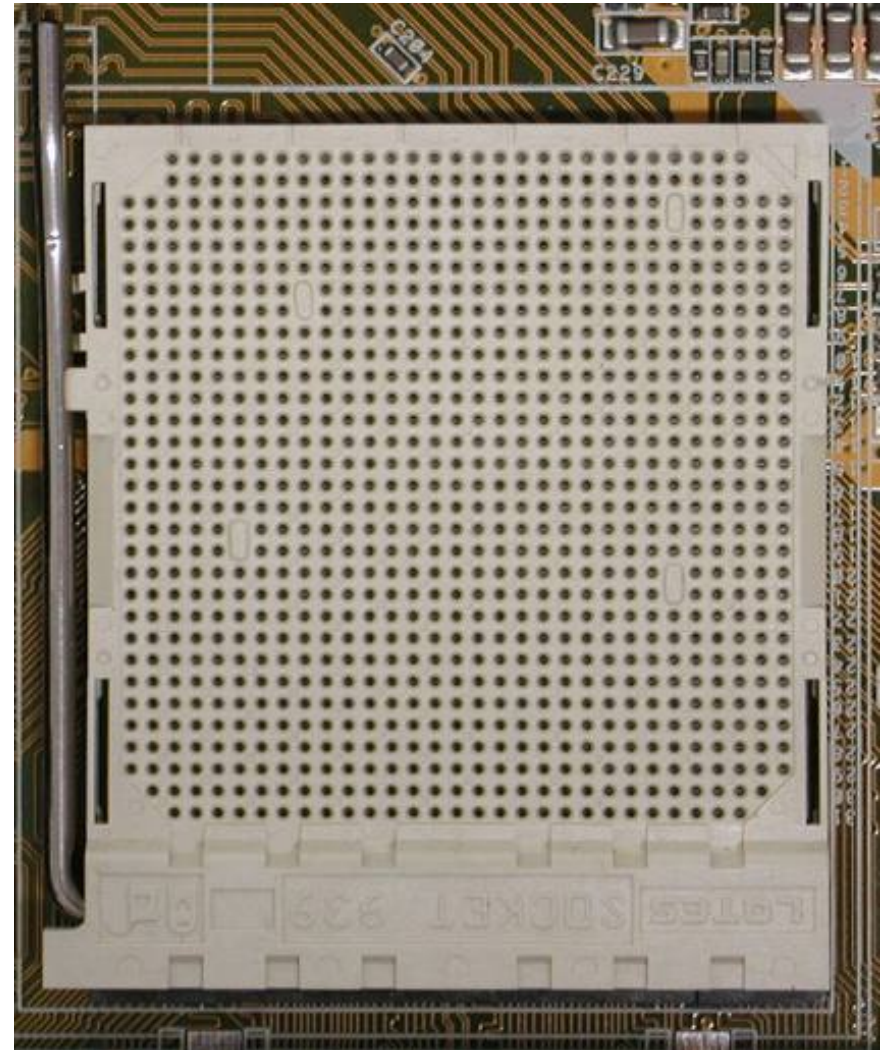
# Плагины



- Хост:
  - Ну и где они все?
  - Ну и что с ними делать?
- Плагины:
  - А что здесь есть?
  - А что мне можно?

# Хост: ну и где они все? (discovering)

- Где-то в `sys.path`
- Просто лежат вокруг
- Какой-нибудь  
спецформат файлов

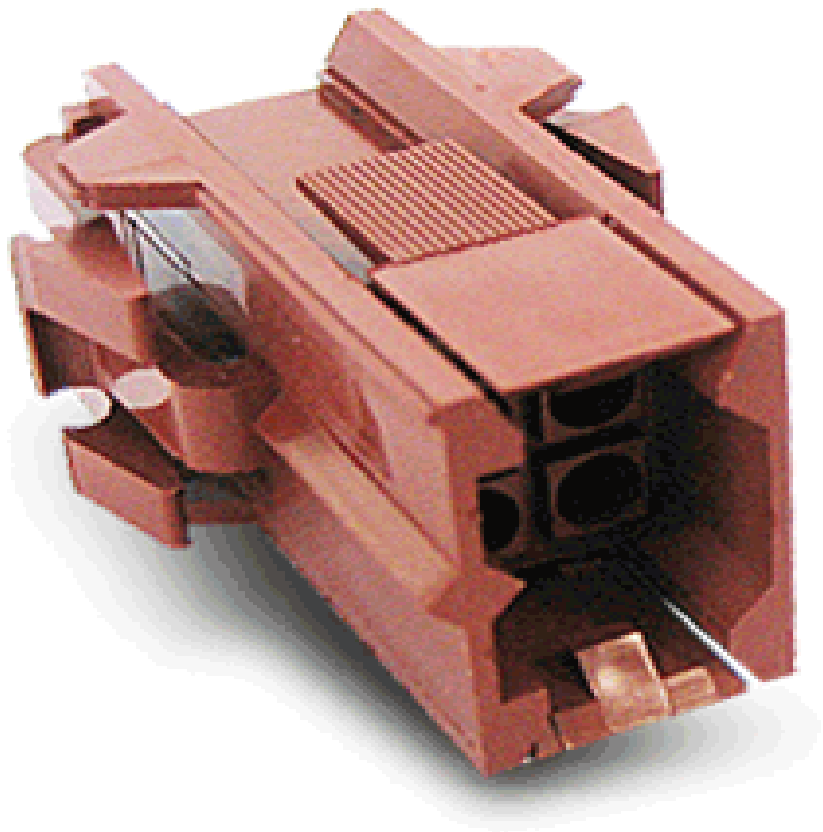


# Хост: ну и что с ними делать? (introspection)

- Конфиг-файлы с метаданными
- Аннотации в коде
- Метаданные в спецформате файлов



# Плагины: а что здесь есть? (services)



- Документация
- Что-нибудь более исполняемое
  - Декларации интерфейсов
  - Тесты

# Плагины: а что мне можно? (extension points)



- Документация
- Что-нибудь более исполняемое
  - Декларации интерфейсов
  - Тесты
- А можно у меня тоже будут плагины?

# Интерфейсы??

интерфейсы в Python  
≠  
интерфейсы в Java



# Интерфейсы как документация

- «должен иметь `__unicode__`»
- «при вызове `foo(n, m)` должен возвращать нечто, что работает как массив»
- ...

```
class IFunctionality(Interface):  
    def __unicode__():  
        """Should blah blah..."""  
    def foo(n, m):  
        """Should blah blah blah..."""  
    ...
```



# Реестр реализаций

```
for p in INavigationPane.implementations():  
    p.contribute_to(navigation_bar)
```

# Мудрость человечества

Расширения Mercurial

ZCA

Плагины Twisted

Плагины Trac



# Mercurial

```
def hello(ui, repo, **opts):  
    """'Hello world' plugin"""  
    print("Hello world!")  
  
cmdtable = {  
    "hello": (hello, [], "hg hello")  
}
```

- Обнаружение плагинов — `/etc/hgrc`, `~/.hgrc`
- Интерфейс — модуль с магическими именами:  
`cmdtable`, `extsetup`, `uisetup`
- Сервисы — весь codebase
- Точки расширения — новые команды, monkey patching чего угодно



# zope.interfaces

```
class IStuff(Interface):  
    def doStuff(foo):  
        """Let's do stuff"""  
  
...  
class MyStuff(object):  
    implements(IStuff)  
    def doStuff(self, foo):  
        ...
```

- Интерфейсы — документация

# ZCA

```
class IStuff(Interface):  
    ...  
    adapts(IRubbish)  
    invariant(check_stuff)  
    ...  
createObject(...)
```

- Помимо документации:
  - фабрики
  - адаптеры
  - проверка инвариантов — исполняемая валидация контрактов
  - etc



# Плагины Twisted

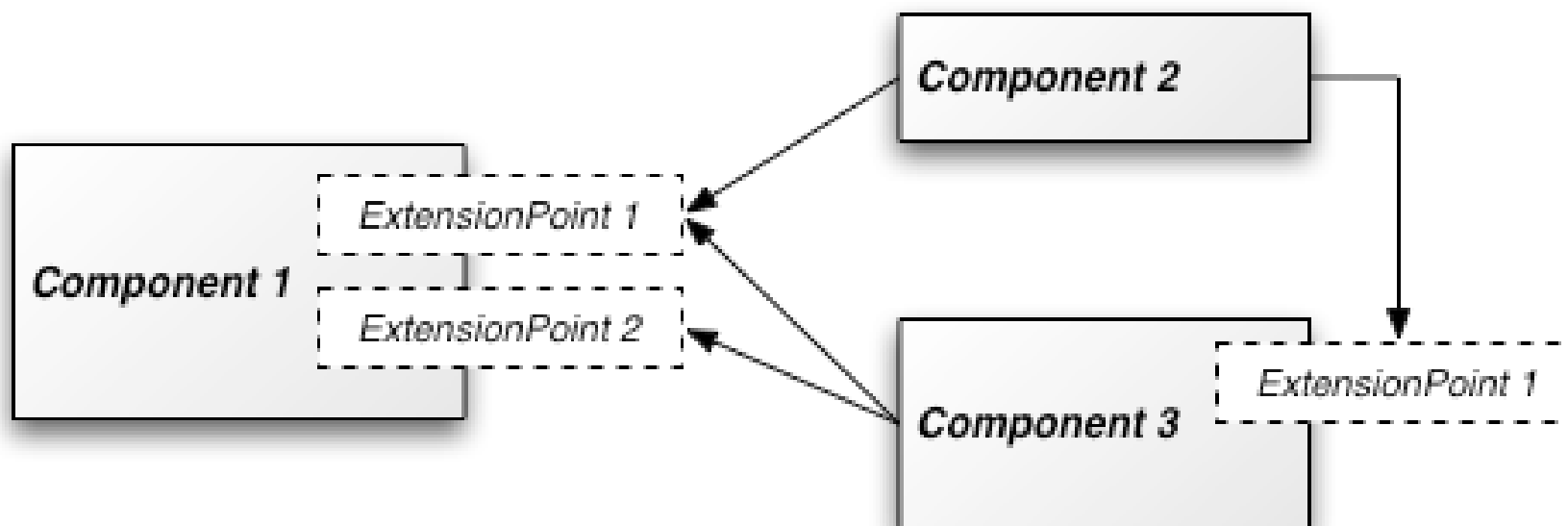
```
from zope import interface
class Feature(interface.IPlugin,
               IFunctionality):
    ...

for i in interface.getPlugins(IFunctionality):
    ...
```

- zope.interfaces
- Java-like наследование
- «Чистый» реестр реализаций



# Плагины Trac



- Компонент — отдельная сущность
- Интерфейсы — типы точек расширения
- Интерфейсы реализуются компонентами



# Trac: интерфейсы

```
from trac.core import *
```

```
class IWikiMacroProvider(Interface):  
    def macros():  
        """Returns macros to be available in wiki"""  
  
class MyComponentWithNewMacros(Component):  
    implements(IWikiMacroProvider)
```

- Декларация и реализация интерфейсов — как в `zope.interfaces`, но относится к компонентам





# Trac: точки расширения

```
from trac.core import *

class WikiComponent(Component):
    macros_providers =
        ExtensionPoint(IWikiMacrosProviders)
    ...
    def render(...):
        macros = [p.macros() for p in
                  self.macros_providers]
```

- Точка расширения — реестр реализаций
- Интерфейс — **не** реестр реализаций
- Переиспользование интерфейсов(!)



# Trac: плагины

▼ **TracWikiRename 2.0** Uninstall

Add simple support for renaming/moving wiki pages

Author: [coderanger@yahoo.com](mailto:coderanger@yahoo.com)  
Home page: <http://trac-hacks.org/wiki/WikiRenamePlugin>  
License: BSD

Component	Enabled
WikiRenameCtxtnavModule wikirename.ctxtnav	<input type="checkbox"/>
WikiRenameModule wikirename.web_ui	<input type="checkbox"/>

Apply changes

- Плагин — контейнер для компонентов



# Trac: упаковка плагинов

```
from setuptools import find_packages, setup
```

```
setup(  
    name='MyComponentWithNewMacros',  
    version='1.0',  
    packages=find_packages(),  
    entry_points = """  
        [trac.plugins]  
        blah = mycomp.component  
    """  
)
```

- egg
- Просто перечислите модули
- Название entry point не играет значения



# Trac: итог

- Поиск плагинов — `setuptools`, `eggs`
- Интроспекция — компоненты
- Сервисы — интерфейсы точек расширения
- Точки расширения — на уровне компонентов

# Trac: переиспользование



- **trac.core** — 220 строк, нет внешних зависимостей
- **trac.loader** — 110 строк, зависимость только на **setuptools**
- 3-BSD лицензия

# Вопросы?

<http://dottedmag.net/talks/>