주피터: 파이썬 노트북
파이썬을 넘어

김대권

주피터: 파이썬 노트북
파이썬을 넘어

SMART STUDY
김대권

PYCON APAC 2016

주피터: 파이썬 노트북
파이썬을 넘어

@nacyo_t

@nacyo_t

Write the Docs

Seoul Meetup #1

강남역 메리츠타워 21층 아카마이 코리아

8월 15일(월요일) 오전 10시 ~ 2시

주피터: 파이썬 노트북, 파이썬을 넘어 @nacyo_t

IPython

IPython Notebook

Jupyter

Jupyter Notebook

IPython

IPython Notebook

Jupyter의 역사

Jupyter Notebook

IPython

IPython Notebook

Jupyter의 역사

Jupyter Notebook

범용성

매체성

IPython

IPython Notebook

Jupyter의 역사

Jupyter Notebook

범용성

매체성

IPython

IPython Notebook

이름

Jupyter의 역사

Jupyter Notebook

IPython

IPython Notebook

Jupyter의 역사

Jupyter Notebook

카오스

IPython

IPython Notebook

Jupyter

Jupyter Notebook

IPython

IPython Notebook

Jupyter

Jupyter Notebook

IPython

IPython Notebook

Jupyter

Jupyter Notebook

웹 인터페이스

IPython

IPython Notebook

웹 인터페이스

Jupyter

Jupyter Notebook

IPython

IPython Notebook

Jupyter

Jupyter Notebook

커맨드 라인

IPython

IPython Notebook

Jupyter

Jupyter Notebook

Python

IPython

IPython Notebook

Jupyter

Jupyter Notebook

```
➜  ~ python
Python 3.5.2 (default, Jul 27 2016, 15:55:28)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>> def hello(name):
...     print("Hello, {}.".format(name))
...
>>> hello('python')
Hello, python.
>>>
```

```
➜  ~ ipython
Python 3.5.2 (default, Jul 27 2016, 15:55:28)
Type "copyright", "credits" or "license" for more information.

IPython 5.0.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print('Hello, world!')
Hello, world!

In [2]: def hello(name):
   ...:     print("Hello, {}.".format(name))
   ...:

In [3]: hello('Jupyter')
Hello, Jupyter.

In [4]:
```

```
➜  ~ python
Python 3.5.2 (default, Jul 27 2016, 15:55:28)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>> def hello(name):
...     print("Hello, {}.".format(name))
...
>>> hello('python')
Hello, python.
>>>
```

R E P L

Read Eval Print Loop

Read Eval Print Loop

Lisp code

```
(loop (print (eval (read))))
```

Lisp code

```
(loop (print (eval (read))))
```

> When you start a Lisp system, it enters a read-eval-print loop. Most other languages have nothing comparable to `read', nothing comparable to `eval', and nothing comparable to `print'. What gaping deficiencies!
>
> — Richard Stallman

> Lisp을 시작하면 read-eval-print-loop를 사용하게 된다. 다른 대부분의 언어들은 `read`에 대응하는 것이 없고, `eval`에 대응하는 것이 없고, `print`에 대응하는 것이 없다. 이 얼마나 엄청난 결함인가.
>
> — Richard Stallman

Read Eval Print Loop

아이디어

Read Eval Print Loop

Lisp, Ruby, Python, Javascript, Clojure, Haskell, …

Read Eval Print Loop

Maple, Mathematica, Statistics Package

Python

IPython을 만든 이유

IPython Notebook

Jupyter

Jupyter Notebook

```
➜  ~ python
Python 3.5.2 (default, Jul 27 2016, 15:55:28)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>> def hello(name):
...     print("Hello, {}.".format(name))
...
>>> hello('python')
Hello, python.
>>>
```

Python

IPython을 만든 이유

IPython Notebook

Jupyter

Jupyter Notebook

```
➜  ~ ipython
Python 3.5.2 (default, Jul 27 2016, 15:55:28)
Type "copyright", "credits" or "license" for more information.

IPython 5.0.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: print('Hello, world!')
Hello, world!

In [2]: def hello(name):
   ...:     print("Hello, {}.".format(name))
   ...:

In [3]: hello('Jupyter')
Hello, Jupyter.

In [4]:
```

> I started using Python in 2001 and liked the language, but its interactive prompt felt like a crippled toy compared to the systems mentioned(maple, mathematica, etc) above or to a Unix shell.
>
> — Fernando Perez

> 나는 2001년부터 파이썬을 사용했고, 파이썬을 좋아하게 되었다. 하지만 인터렉티브 프롬프트는 Maple, Mathematica, 유닉스 셸에 비하면 망가진 장난감 같았다.
>
> — Fernando Perez

IPython 0.0.1

2001년

$PYTHONSTARTUP 스크립트

Python 기본 REPL의 확장

리서치를 위한 헬퍼 함수들

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

IPython 0.0.1~

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

Python REPL 확장으로 발전

IPython 0.12

2011년 12월 (0.0.1로부터 10년)

IPython Notebook의 등장

0.0.1	2001
0.11	2011
0.12	**2011**
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

“

The major new feature with this release is the IPython Notebook, an interactive Python interface running in the browser.

— IPython Release 0.12

0.0.1 2001
0.11 2011
0.12 2011
0.13 2012
1.0 2013
2.0 2014
3.0 2015
4.0 2015
5.0 2015

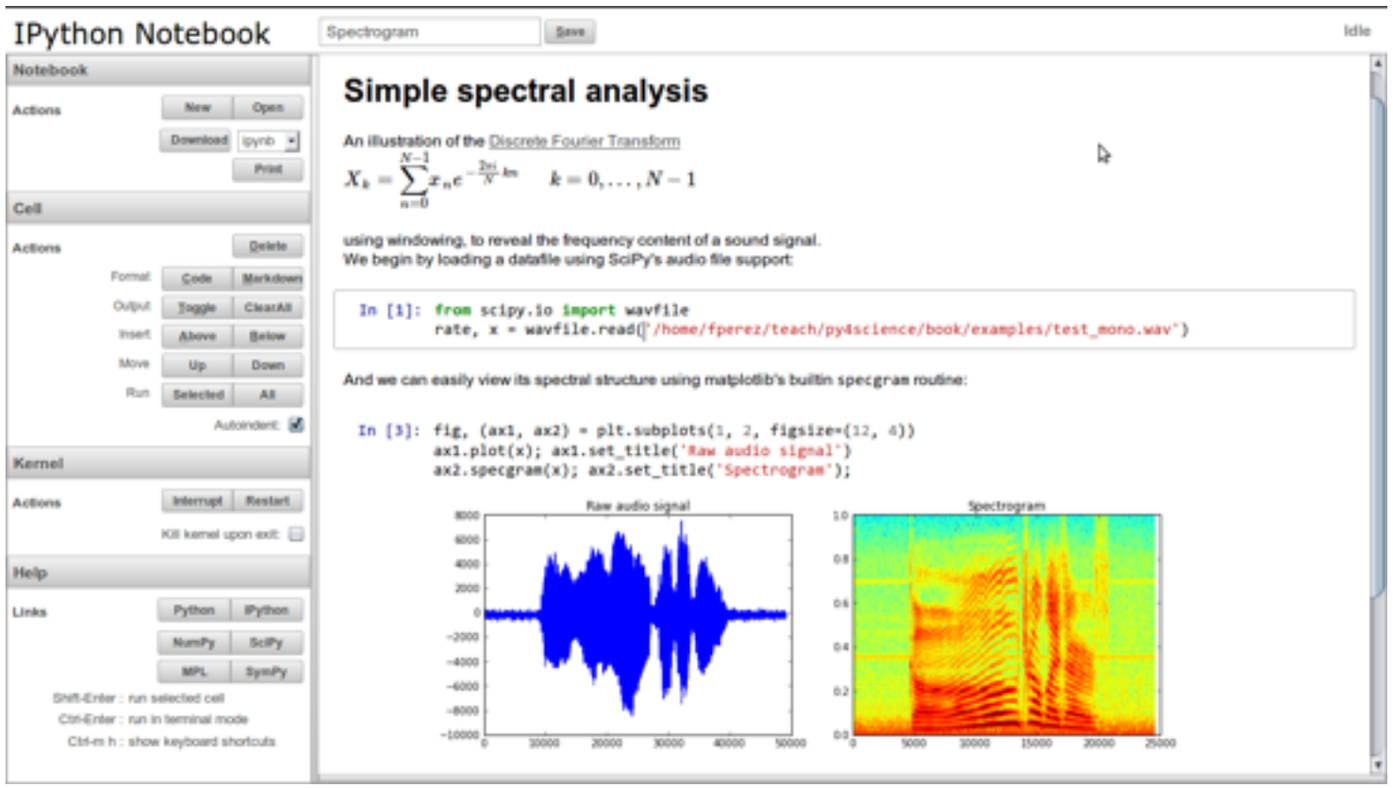
“

이번 릴리즈의 중요한 새로운 기능은 브라우저 환경에서 작동하는 인터렉티브 인터페이스인 IPython Notebook입니다.

— IPython Release 0.12

0.0.1 2001
0.11 2011
0.12 2011
0.13 2012
1.0 2013
2.0 2014
3.0 2015
4.0 2015
5.0 2015

IPython Notebook



0.0.1 2001

0.11 2011

0.12 2011

0.13 2012

1.0 2013

2.0 2014

3.0 2015

4.0 2015

5.0 2015

IPython Notebook

웹 인터페이스 기반 REPL

셀 단위 편집 지원

Markdown 셀 지원

JSON 포맷의 문서

PDF, HTML 출력 가능

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

IPython 0.12


100
75
50
25
Jan 1, 2004
Feb 1, 2009
Mar 1, 2014
Dec 2011
IPython 23
IPython notebook 1

IPython 0.12

IPython Notebook이 만들어진 기반

0.0.1	2001
0.11	2011
0.12	**2011**
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

IPython 0.12

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

앞으로

IPython 0.11

0.0.1	2001
0.11	**2011**
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

파이썬 기반의 단일 프로그램

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

IPython 0.11

PyZMQ 개발 및 ZeroMQ 도입

Qt 콘솔

Python3 Support

Git / Github로 이전

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

> This is where early 2010 found us, and then zerendipity struck: while on a month-long teaching trip to Colombia I read an article about ZeroMQ and talked to Brian about it, as it seemed to provide the right abstractions for us with a simpler model than Twisted. Brian then blew me away, coming back in two days with a new set of clean Cython-based bindings: we now had pyzmq!
>
> — Fernando Perez

> “ 2010년 초에 우연한 만남에서 비롯된다. 내가 교육을 위해 한달 간 콜롬비아 여행을 하는 동안 ZeroMQ에 관한 글을 읽고 Brian에게 ZeroMQ가 Twisted보다 단순한 모델로 올바른 추상화 레이어가 될 것 같다고 이야기했다. Brian은 이틀만에 ZeroMQ의 Cython 기반 바인딩을 개발해왔다. pyzmq가 개발된 것이다.
>
> — Fernando Perez

> It was the perfect blend of pair programming and simultaneous development, and in just two days we had a prototype of a python shell over zmq working, (중략) so I've posted it for reference as a standalone github repository
>
> — Fernando Perez

> 마치 페어 프로그래밍과 동시적 개발을 섞어놓은 것 같았다. 단 이틀만에 zmq 기반의 우리는 동작하는 파이썬 셸 프로토타입을 만들었다. (중략) 나는 이 때 개발한 소스 코드를 레퍼런스로 활용할 수 있도록 github 에 저장소에 올려두었다.

— Fernando Perez

fperez/zmq-pykernel

frontend.py

kernel.py

message_spec.rst

0.0.1 2001

0.11 2011

0.12 2011

0.13 2012

1.0 2013

2.0 2014

3.0 2015

4.0 2015

5.0 2015

```
➜ zmq-pykernel git:(master) ✗ ./kernel.py
Starting the kernel...
On: tcp://127.0.0.1:5555 tcp://127.0.0.1:5556
Use Ctrl-\ (NOT Ctrl-C!) to terminate.
```

```
➜ zmq-pykernel git:(master) ✗
```

```
➜ zmq-pykernel git:(master) ✗ ./kernel.py
Starting the kernel...
On: tcp://127.0.0.1:5555 tcp://127.0.0.1:5556
Use Ctrl-\ (NOT Ctrl-C!) to terminate.
```

```
➜ zmq-pykernel git:(master) ✗ ./frontend.py
Python 2.7.12 (default, Jul 20 2016, 17:14:16)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(Console)
Py>>>
```

```
➜ zmq-pykernel git:(master) ✗ ./kernel.py
Starting the kernel...
On: tcp://127.0.0.1:5555 tcp://127.0.0.1:5556
Use Ctrl-\ (NOT Ctrl-C!) to terminate.
{u'content': {u'code': u"print('Hello, IPython kernel!')"},
 u'header': {u'username': u'toto', u'msg_id': 0, u'session': u'0eae1064-8152
-4bbb-b2c7-024e786f6ca4'},
 u'msg_type': u'execute_request',
 u'parent_header': {}}
{'content': {u'data': 'Hello, IPython kernel!\n', u'name': u'stdout'},
 'header': {'username': u'kernel', 'msg_id': 1, 'session': 'bc240247-3254-42
08-9d94-8f34ae28b0e8'},
 'msg_type': u'stream',
 'parent_header': {}}
{'content': {'status': 'ok'},
 'header': {'username': u'kernel', 'msg_id': 2, 'session': 'bc240247-3254-42
08-9d94-8f34ae28b0e8'},
 'msg_type': u'execute_reply',
 'parent_header': {u'username': u'toto', u'msg_id': 0, u'session': u'0eae106
4-8152-4bbb-b2c7-024e786f6ca4'}}
```

```
➜ zmq-pykernel git:(master) ✗ ./frontend.py
Python 2.7.12 (default, Jul 20 2016, 17:14:16)
[GCC 4.2.1 Compatible Apple LLVM 7.3.0 (clang-703.0.31)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
(Console)
Py>>> print('Hello, IPython kernel!')
Hello, IPython kernel!
Py>>>
```

```
{u'content': {u'code': u"print('Hello, IPython kernel!')"},
 u'header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'},
 u'msg_type': u'execute_request',
 u'parent_header': {}}

{'content': {u'data': 'Hello, IPython kernel!\n', u'name': u'stdout'},
 'header': {'username': u'kernel', 'msg_id': 1, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'stream',
 'parent_header': {}}

{'content': {'status': 'ok'},
 'header': {'username': u'kernel', 'msg_id': 2, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'execute_reply',
 'parent_header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'}}
```

```
{u'content': {u'code': u"print('Hello, IPython kernel!')"},
 u'header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'},
 u'msg_type': u'execute_request',
 u'parent_header': {}}

{'content': {u'data': 'Hello, IPython kernel!\n', u'name': u'stdout'},
 'header': {'username': u'kernel', 'msg_id': 1, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'stream',
 'parent_header': {}}

{'content': {'status': 'ok'},
 'header': {'username': u'kernel', 'msg_id': 2, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'execute_reply',
 'parent_header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'}}
```

```
{u'content': {u'code': u"print('Hello, IPython kernel!')"},
 u'header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'},
 u'msg_type': u'execute_request',
 u'parent_header': {}}

{'content': {u'data': 'Hello, IPython kernel!\n', u'name': u'stdout'},
 'header': {'username': u'kernel', 'msg_id': 1, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'stream',
 'parent_header': {}}

{'content': {'status': 'ok'},
 'header': {'username': u'kernel', 'msg_id': 2, 'session':
'bc240247-3254-4208-9d94-8f34ae28b0e8'},
 'msg_type': u'execute_reply',
 'parent_header': {u'username': u'toto', u'msg_id': 0, u'session':
u'0eae1064-8152-4bbb-b2c7-024e786f6ca4'}}
```

서버 프로그램

IPython 0.11

PyZMQ 개발 및 ZeroMQ 도입

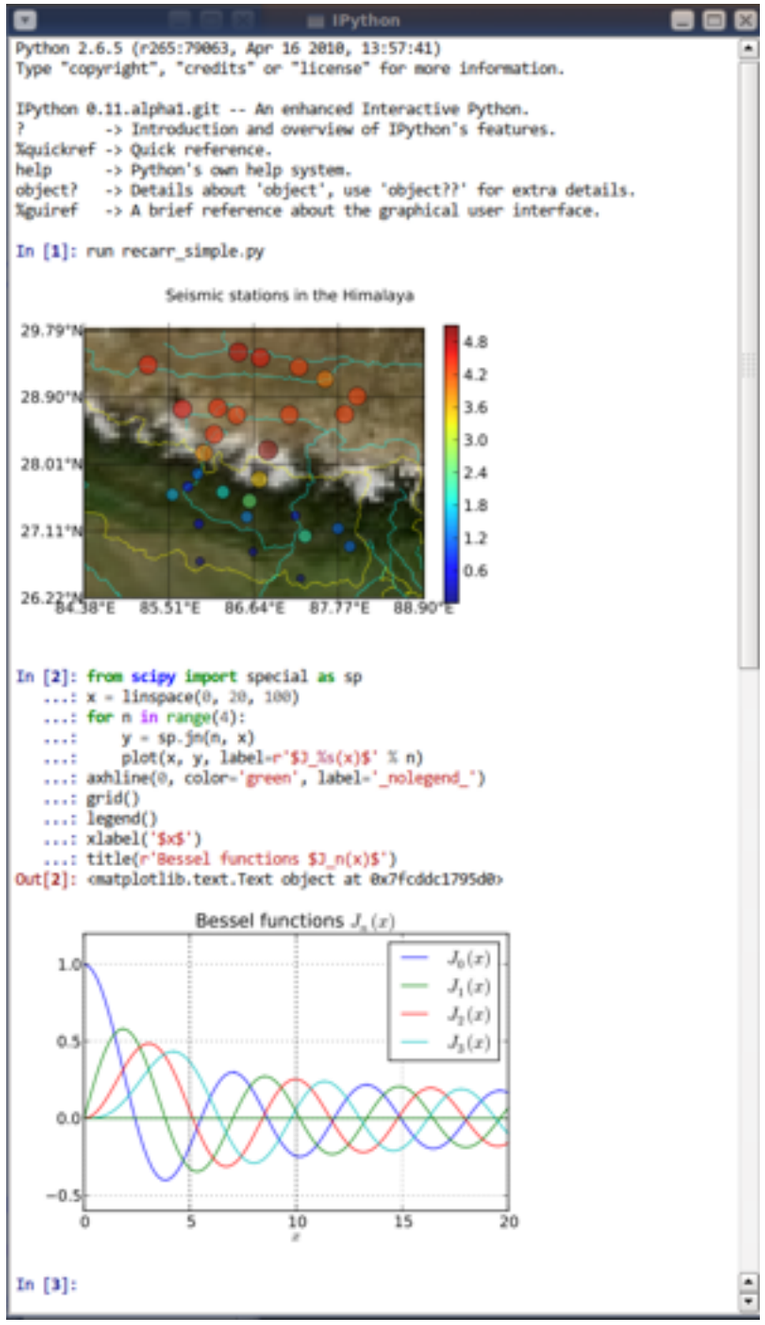
Qt 콘솔

Python3 Support

Git / Github로 이전

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

Qt 콘솔


IPython
Python 2.6.5 (r265:79063, Apr 16 2010, 13:57:41)
Type "copyright", "credits" or "license" for more information.
IPython 0.11.alpha1.git -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
%guiref -> A brief reference about the graphical user interface.
In [1]: run recarr_simple.py
Seismic stations in the Himalaya
29.79°N
28.90°N
28.01°N
27.11°N
26.22°N
84.38°E
85.51°E
86.64°E
87.77°E
88.90°E
4.8
4.2
3.6
3.0
2.4
1.8
1.2
0.6
In [2]: from scipy import special as sp
...: x = linspace(0, 20, 100)
...: for n in range(4):
...: y = sp.jn(n, x)
...: plot(x, y, label=r'$J_%s(x)$' % n)
...: axhline(0, color='green', label='_nolegend_')
...: grid()
...: legend()
...: xlabel('x')
...: title(r'Bessel functions $J_n(x)$')
Out[2]: <matplotlib.text.Text object at 0x7fcddc1795d0>
Bessel functions $J_n(x)$
$J_0(x)$
$J_1(x)$
$J_2(x)$
$J_3(x)$
1.0
0.5
0.0
−0.5
0
5
10
15
20
x
In [3]:

> There is a new GUI framework for IPython, based on a client-server model in which multiple clients can communicate with one IPython kernel, using the ZeroMQ messaging framework. There is already a Qt console client, which can be started by calling ipython qtconsole.
>
> — IPython Release 0.11

> IPython에서 ZeroMQ 메시징 프레임워크를 기반으로 하나의 IPython 커널과 다수의 클라이언트가 통신할 수 있는 클라이언트-서버 모델에 기반한 새로운 GUI 프레임워크를 지원합니다. `ipython qtconsole` 명령어로 Qt 콘솔을 실행할 수 있습니다.
>
> — IPython Release 0.11

ZeroMQ 도입

커널과 클라이언트 분리

메시지 프로토콜 작성

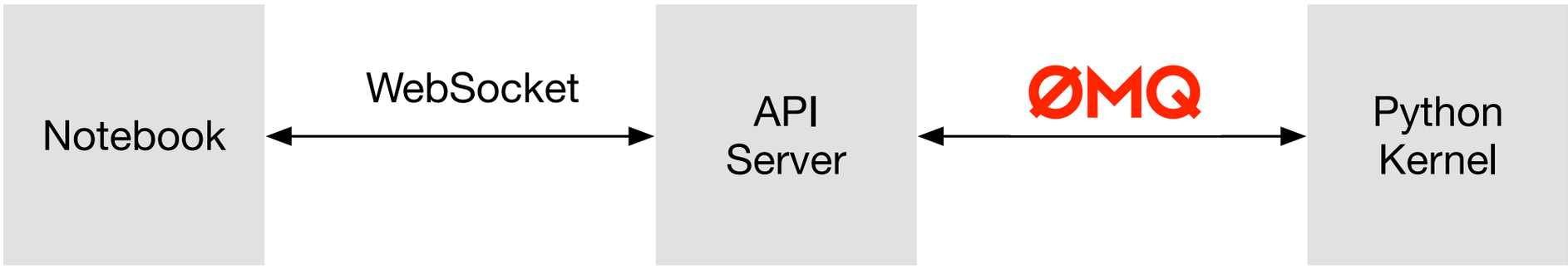
커널은 메시지만 전달할 수 있으면 됨

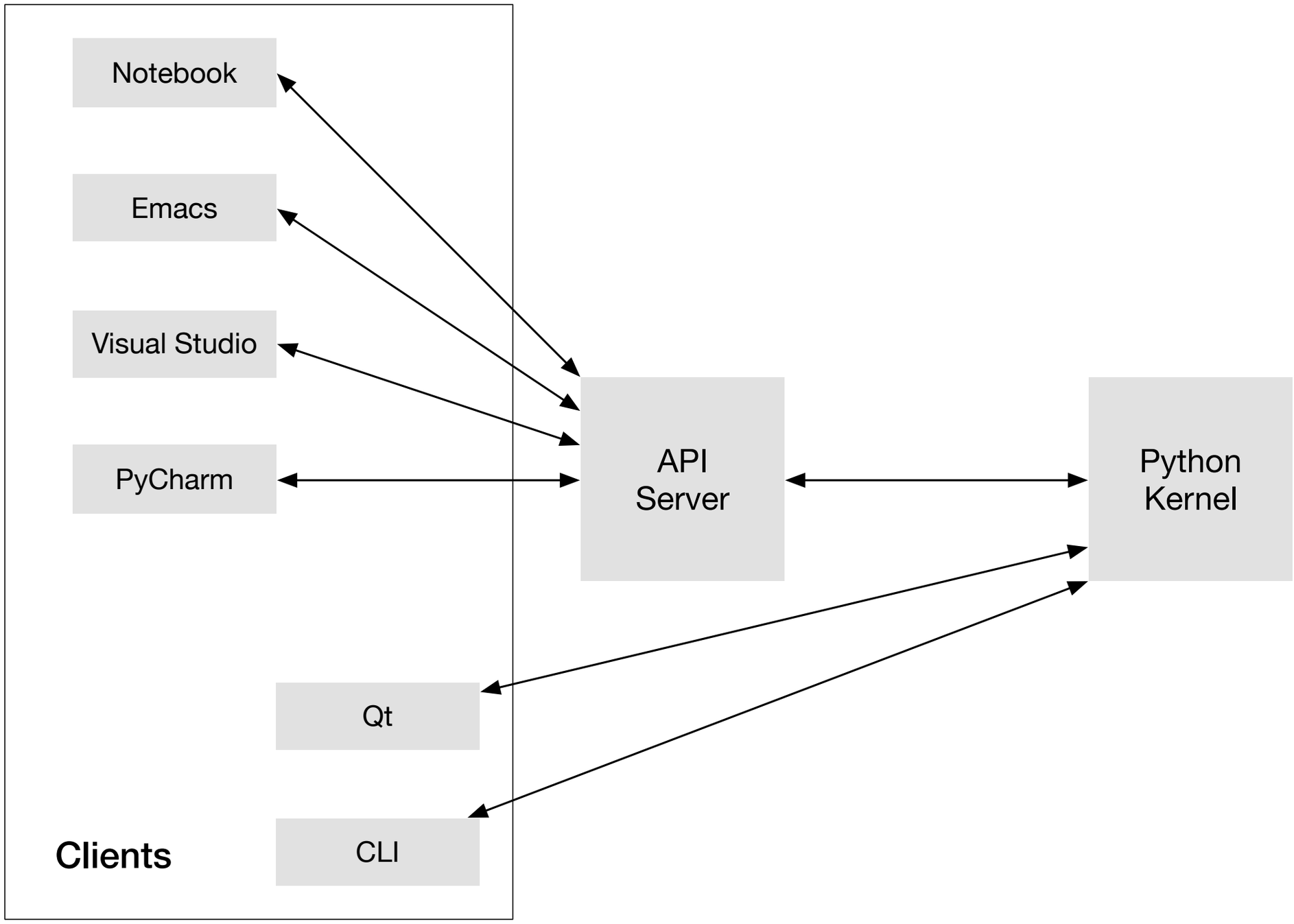
파이썬이 아니라도 가능

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

서버 프로그램

서버 프로그램




Notebook
Emacs
Visual Studio
PyCharm
Qt
CLI
Clients
API
Server
Python
Kernel

ZeroMQ 도입

커널과 클라이언트 분리

메시지 프로토콜 작성

커널은 메시지만 전달할 수 있으면 됨

~~**다양한 커널 등장을 예고**~~

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

The IPython notebook: a historical retrospective

— Fernando Perez

0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

IPython 0.13

0.0.1 2001

0.11 2011

0.12 2011

0.13 2012

1.0 2013

2.0 2014

3.0 2015

4.0 2015

5.0 2015

셀 매직 커맨드 지원(%%)

IPython 0.12 ~ 2.0

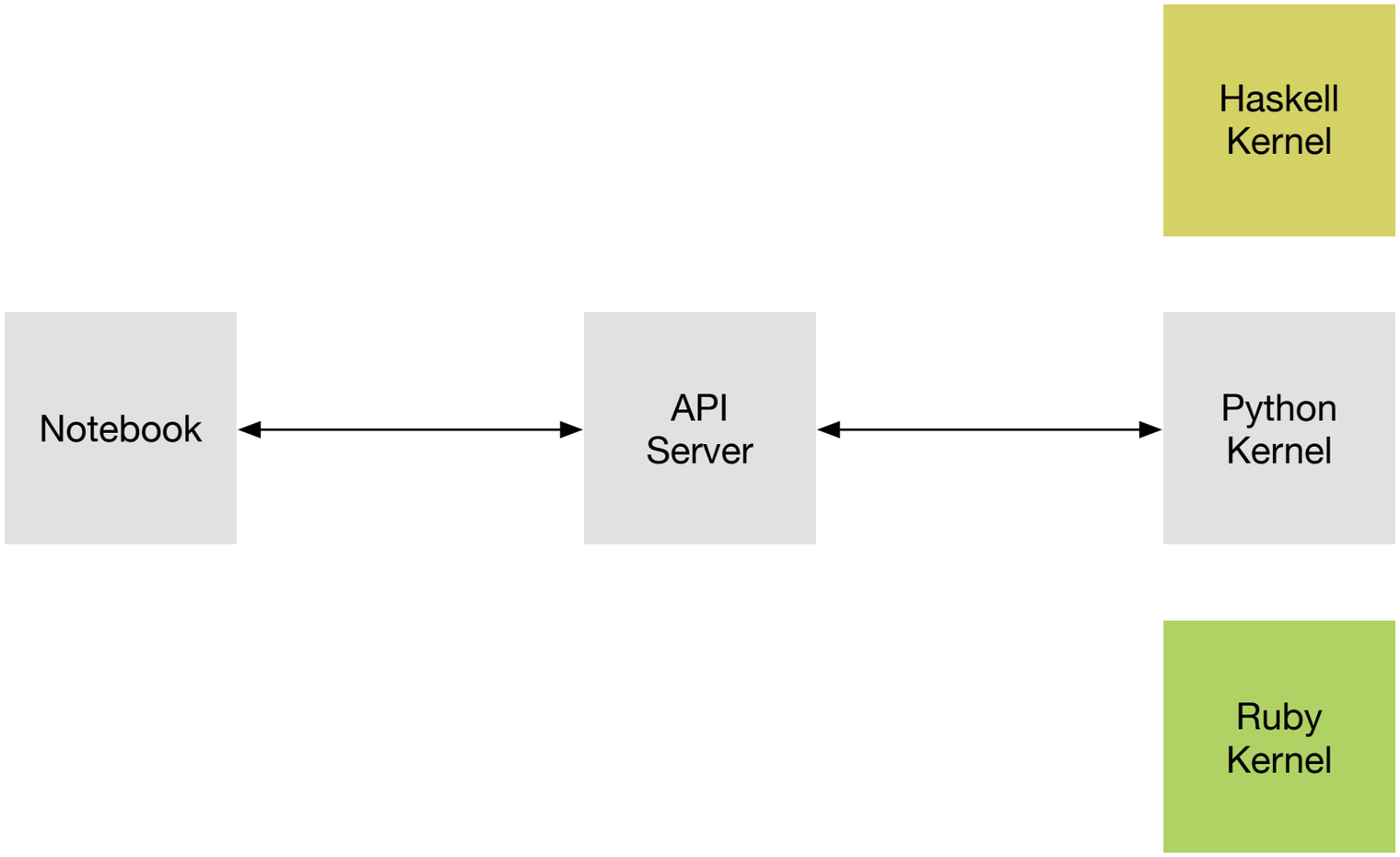
IPython Notebook으로서 정체성을 갖춰나감

파이썬 이외의 커널들이 다수 등장

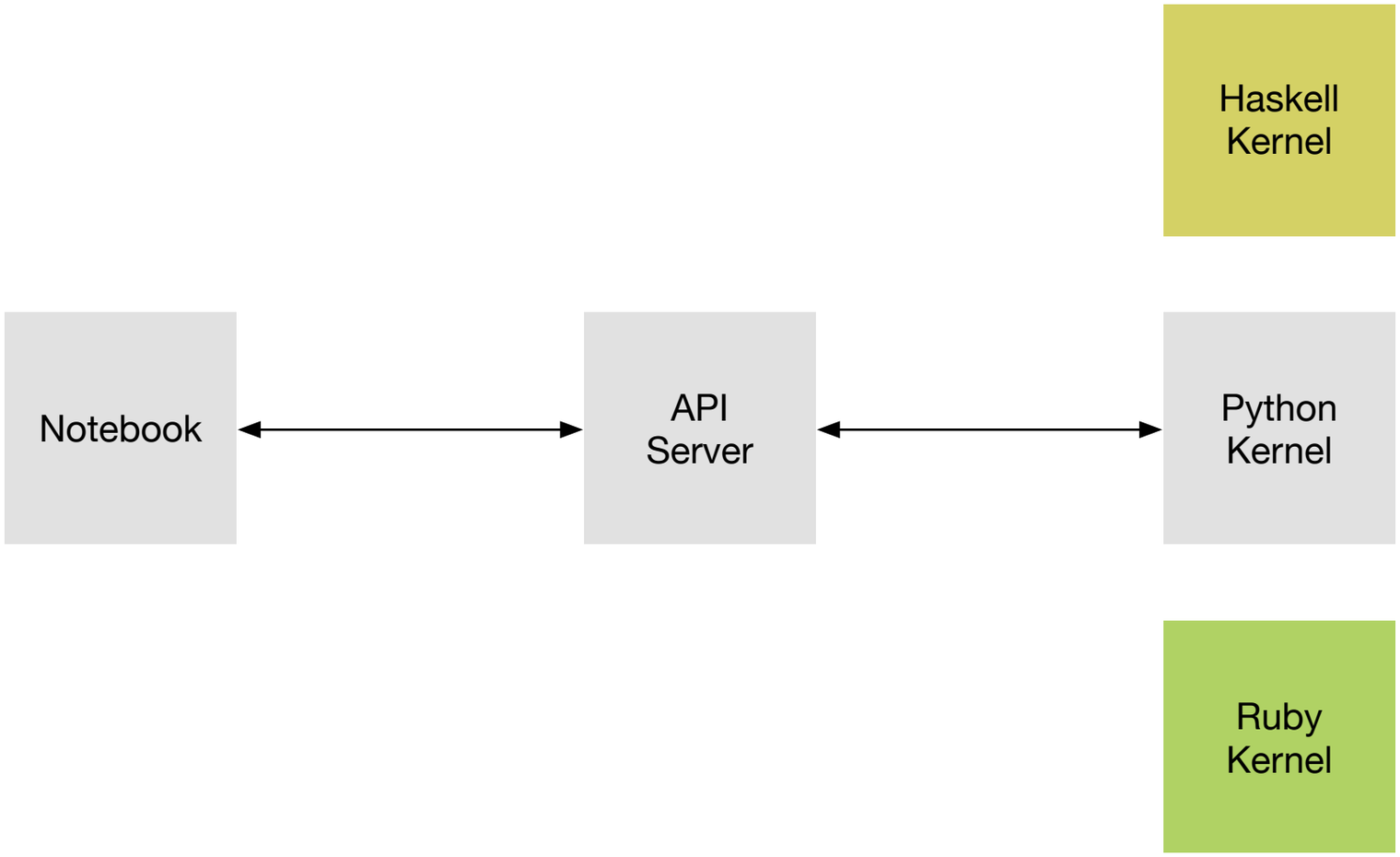
2014년 말 기준: 20개 이상의 커널

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

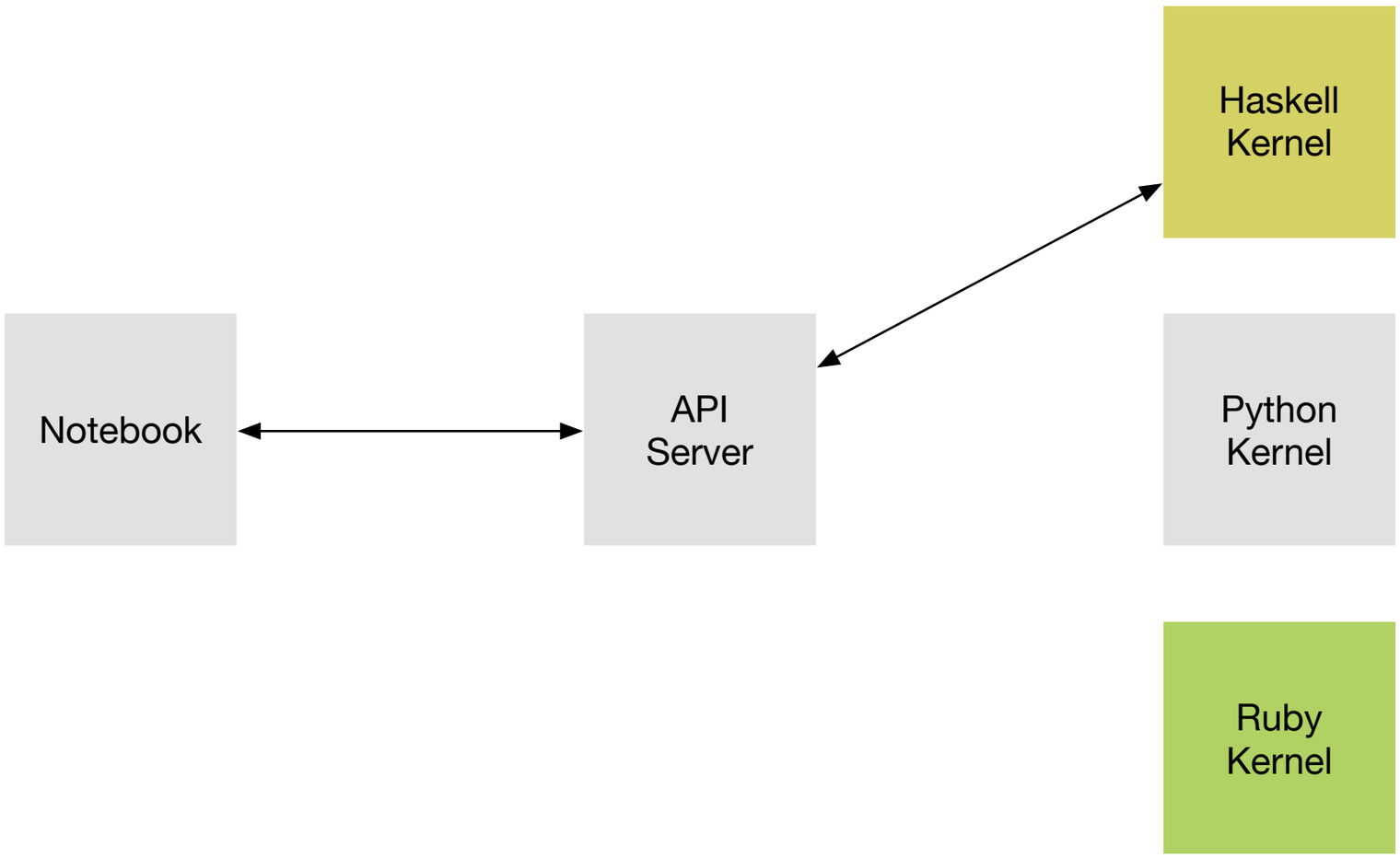
다양한 커널 지원


Haskell
Kernel
Notebook
API
Server
Python
Kernel
Ruby
Kernel

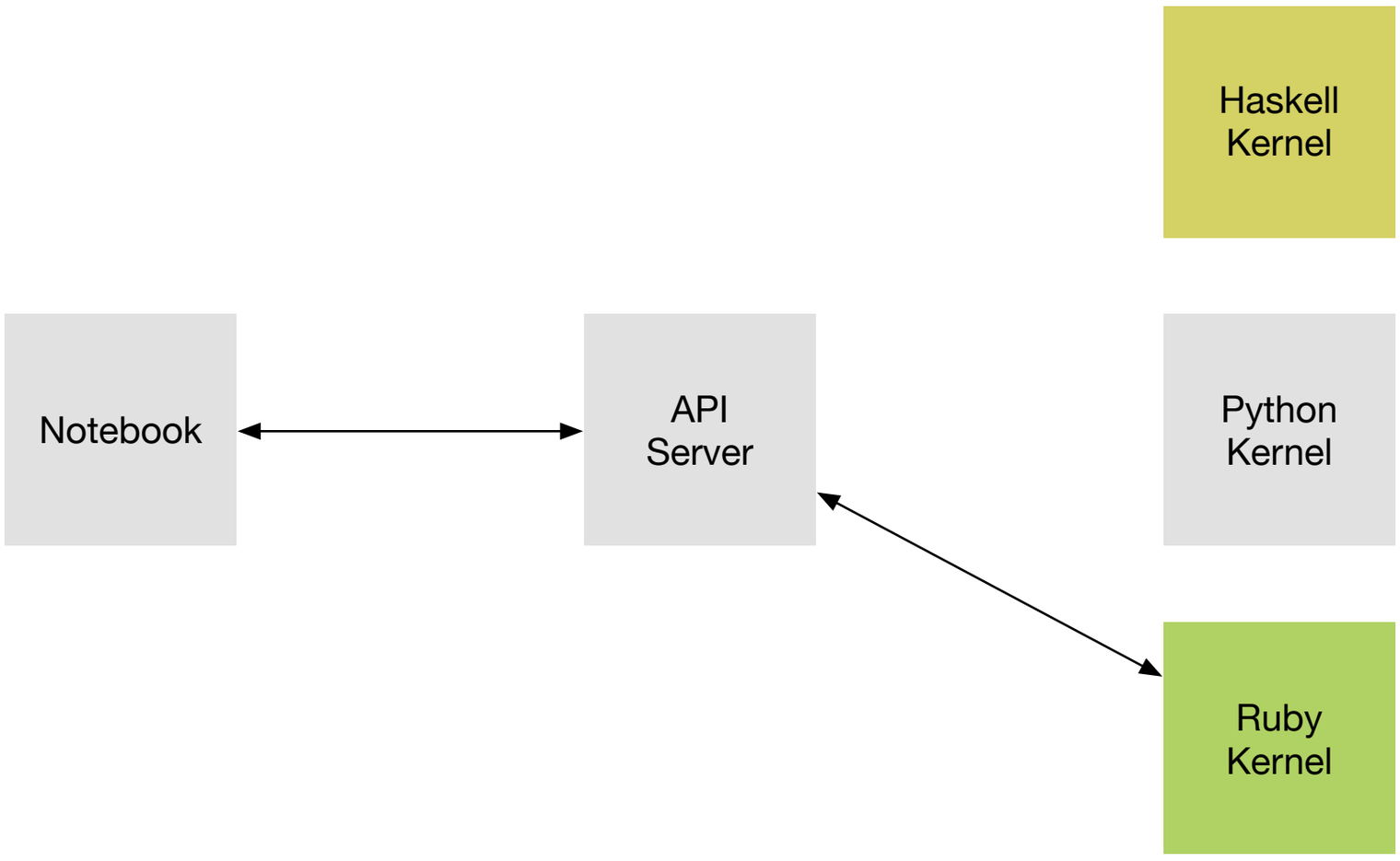
```
$ ipython notebook
```



```
$ ipython notebook --profile=haskell
```



```
$ ipython notebook --profile=ruby
```



IPython 3.0

Jupyter란 이름의 첫 등장

IPython Notebook이 Jupyter 로고를 사용

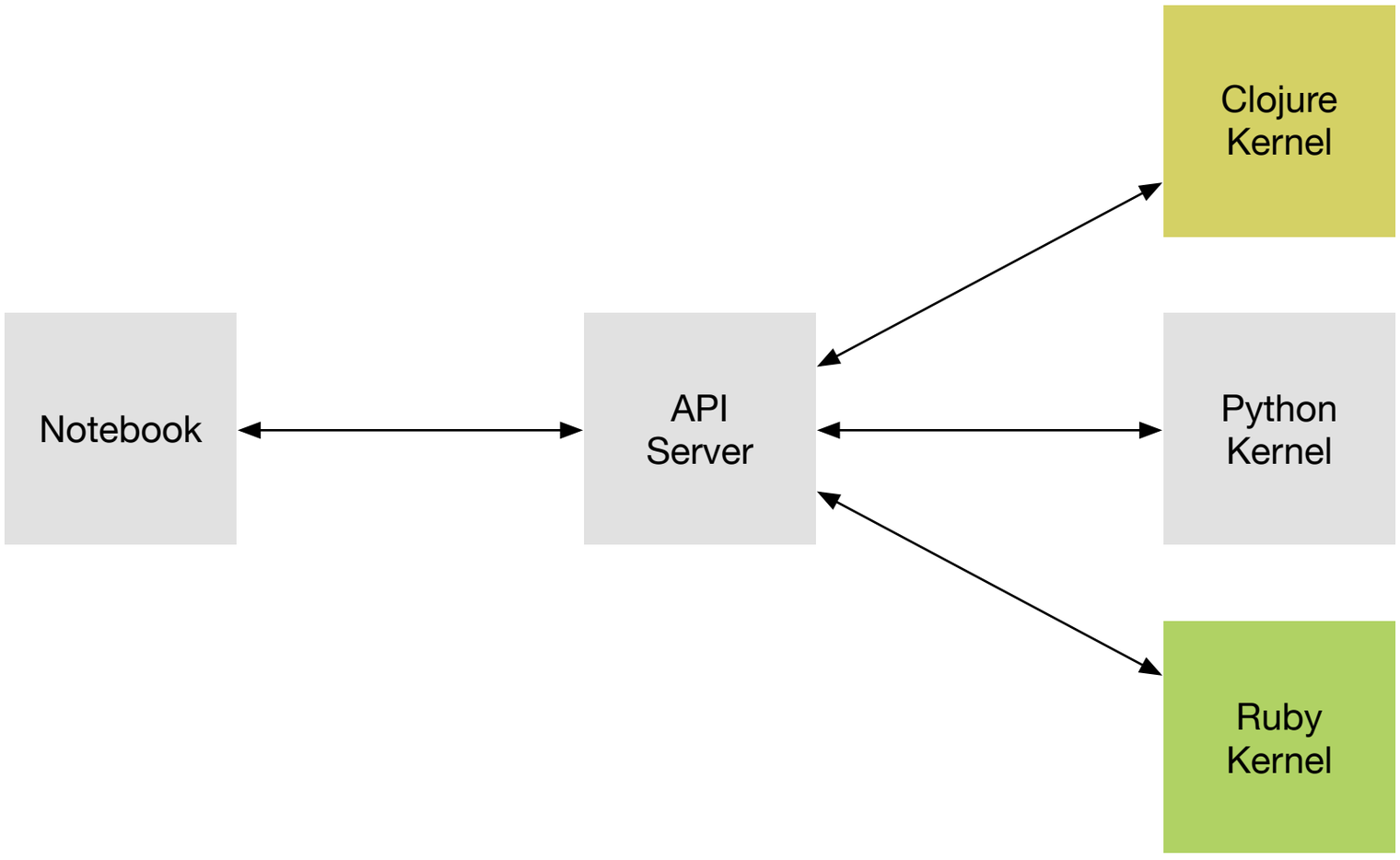
하나의 서버에서 다수의 커널 프로필 지원

새로운 목표

  언어의 의존적이지 않은 부분 분리

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

```
$ jupyter notebook
```



> The 3.x release series will be the last where IPython is released as one big package. For 4.0, we will split up components into several packages. The parts which work for any language will be called Jupyter, while the parts specific to executing Python code will remain as IPython.
>
> — IPython News: IPython 3.0

> IPython이 하나의 거대한 패키지로 릴리즈 되는 것은 3.x 버전이 마지막입니다. 4.0부터는 컴포넌트들이 개별 패키지로 나누어질 것입니다. 언어와 무관하게 작동하는 부분들은 Jupyter로 분리되고, 파이썬 코드를 실행하기 위한 부분만이 IPython으로 남게 됩니다.
>
> — IPython News: IPython 3.0

0.0.1

0.11

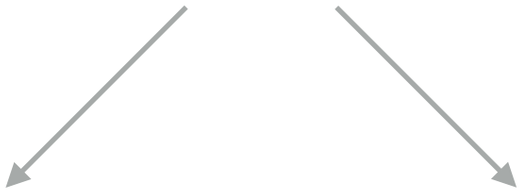
0.12

0.13

1.0

2.0

3.0



IPython 4.0

Notebook 4.0

IPython 5.0

IPython 4.0

IPython에는 Python과 관련된 코드만 남음

Github Jupyter Organization

Notebook

The Big Split™

0.0.1 2001
0.11 2011
0.12 2011
0.13 2012
1.0 2013
2.0 2014
3.0 2015
4.0 2015
5.0 2015

The Big Split™

IPython.utils.traitlets ⇒ traitlets

IPython.html ⇒ notebook

IPython.nbconvert ⇒ nbconvert

IPython.nbformat -> nbformat

IPython.parallel ⇒ ipyparallel

IPython.qt ⇒ qtconsole

IPython.terminal.console ⇒ jupyter_console

IPython.kernel ⇒ jupyter_client

IPython.kernel ⇒ ipykernel

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	**2015**
5.0	2015

The Big Split™

IPython.utils.traitlets ⇒ traitlets

IPython.html ⇒ notebook

IPython.nbconvert ⇒ nbconvert

IPython.nbformat -> nbformat

IPython.parallel ⇒ ipyparallel

IPython.qt ⇒ qtconsole

IPython.terminal.console ⇒ jupyter_console

IPython.kernel ⇒ jupyter_client

IPython.kernel ⇒ ipykernel

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	2015
5.0	2015

The Big Split™

IPython.utils.traitlets ⇒ traitlets

IPython.html ⇒ notebook

IPython.nbconvert ⇒ nbconvert

IPython.nbformat -> nbformat

IPython.parallel ⇒ ipyparallel

IPython.qt ⇒ qtconsole

IPython.terminal.console ⇒ jupyter_console

IPython.kernel ⇒ jupyter_client

IPython.kernel ⇒ ipykernel

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	**2015**
5.0	2015

The Big Split™

IPython.utils.traitlets ⇒ traitlets

IPython.html ⇒ notebook

IPython.nbconvert ⇒ nbconvert

IPython.nbformat -> nbformat

IPython.parallel ⇒ ipyparallel

IPython.qt ⇒ qtconsole

IPython.terminal.console ⇒ jupyter_console

IPython.kernel ⇒ jupyter_client

IPython.kernel ⇒ ipykernel

0.0.1	2001
0.11	2011
0.12	2011
0.13	2012
1.0	2013
2.0	2014
3.0	2015
4.0	**2015**
5.0	2015

Jupyter Ascending

Jupyter


100
75
50
25
2004. 1. 1.
2008. 4. 1.
1월 2015
ipython 78
ipython notebook 31
jupyter 1
jupyter notebook 0

Python

IPython

IPython Notebook

Jupyter

Jupyter Notebook

IPython

IPython Notebook

Jupyter

Jupyter Notebook

IPython

~~IPython Notebook~~

Jupyter

Jupyter Notebook

Jupyter Console

```
$ pip install ipython
```

IPython 5.0

~~IPython Notebook~~

Jupyter

Jupyter Notebook 4.2.2

Jupyter Console

커맨드 라인

파이썬

REPL

```
$ pip install notebook
```

IPython 5.0

~~IPython Notebook~~

Jupyter

Jupyter Notebook 4.2.2

Jupyter Console

웹 기반

인터렉티브

노트북

```
$ pip install jupyter-console
```

IPython 5.0

~~IPython Notebook~~

Jupyter

Jupyter Notebook 4.2.2

Jupyter Console 5.0.0

커맨드 라인

주피터 커널

REPL

(IPython 의존)

```
$ pip install jupyter
```

IPython 5.0

~~IPython Notebook~~

Jupyter 1.0.0

Jupyter Notebook 4.2.2

Jupyter Console 5.0.0

메타 패키지

notebook

qtconsole

jupyter-core

jupyter-clinet

jupyter-console

nbformat

nbconvert

ipykernel

ipywidgets

```
$ pip install jupyter
```

IPython 5.0

~~IPython Notebook~~

Jupyter 1.0.0

Jupyter Notebook 4.2.2

Jupyter Console 5.0.0

메타 패키지

notebook

qtconsole

jupyter-core

jupyter-clinet

jupyter-console

nbformat

nbconvert

ipykernel

ipywidgets

IPython 5.0

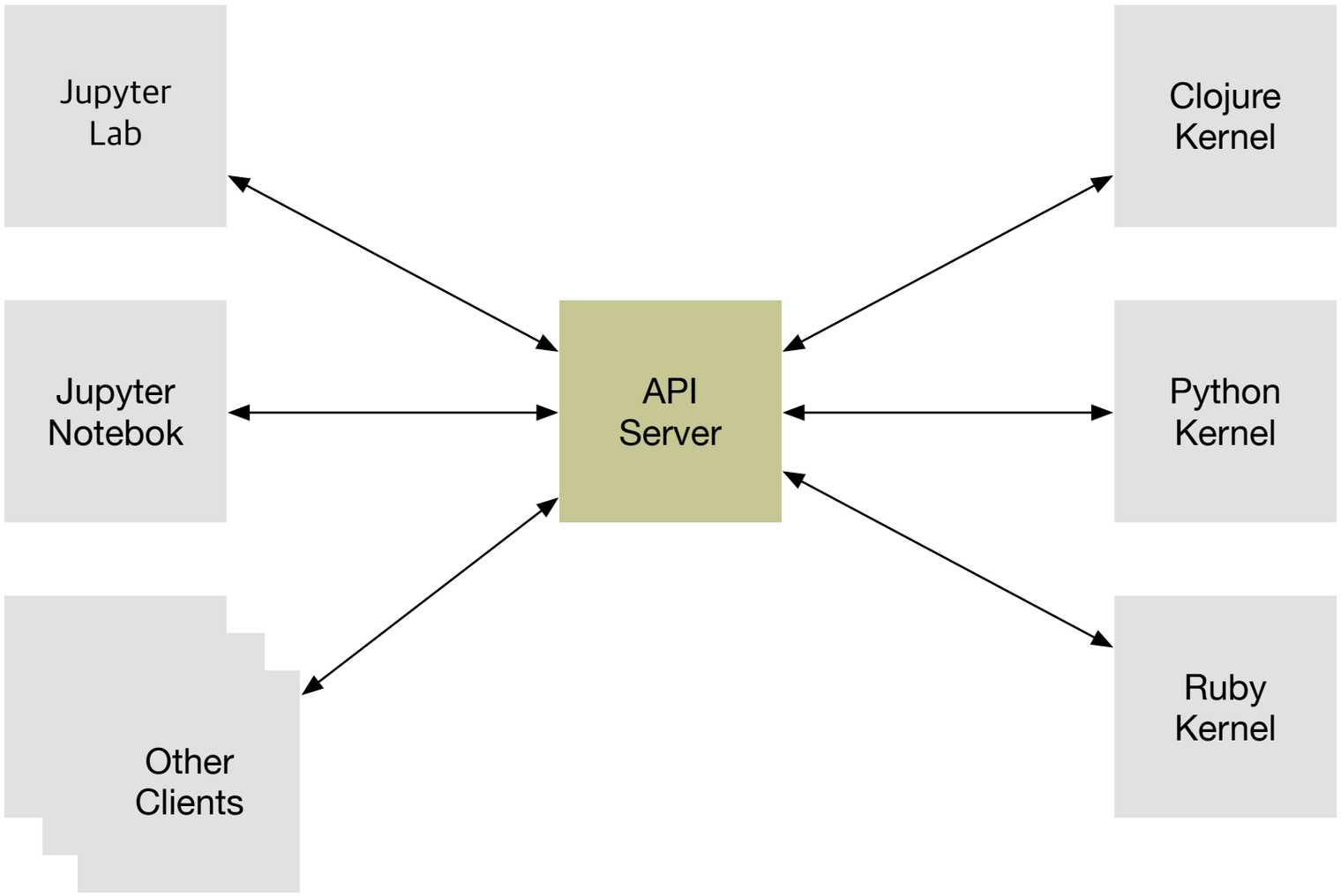
Jupyter 1.0.0

Jupyter Notebook 4.2.2

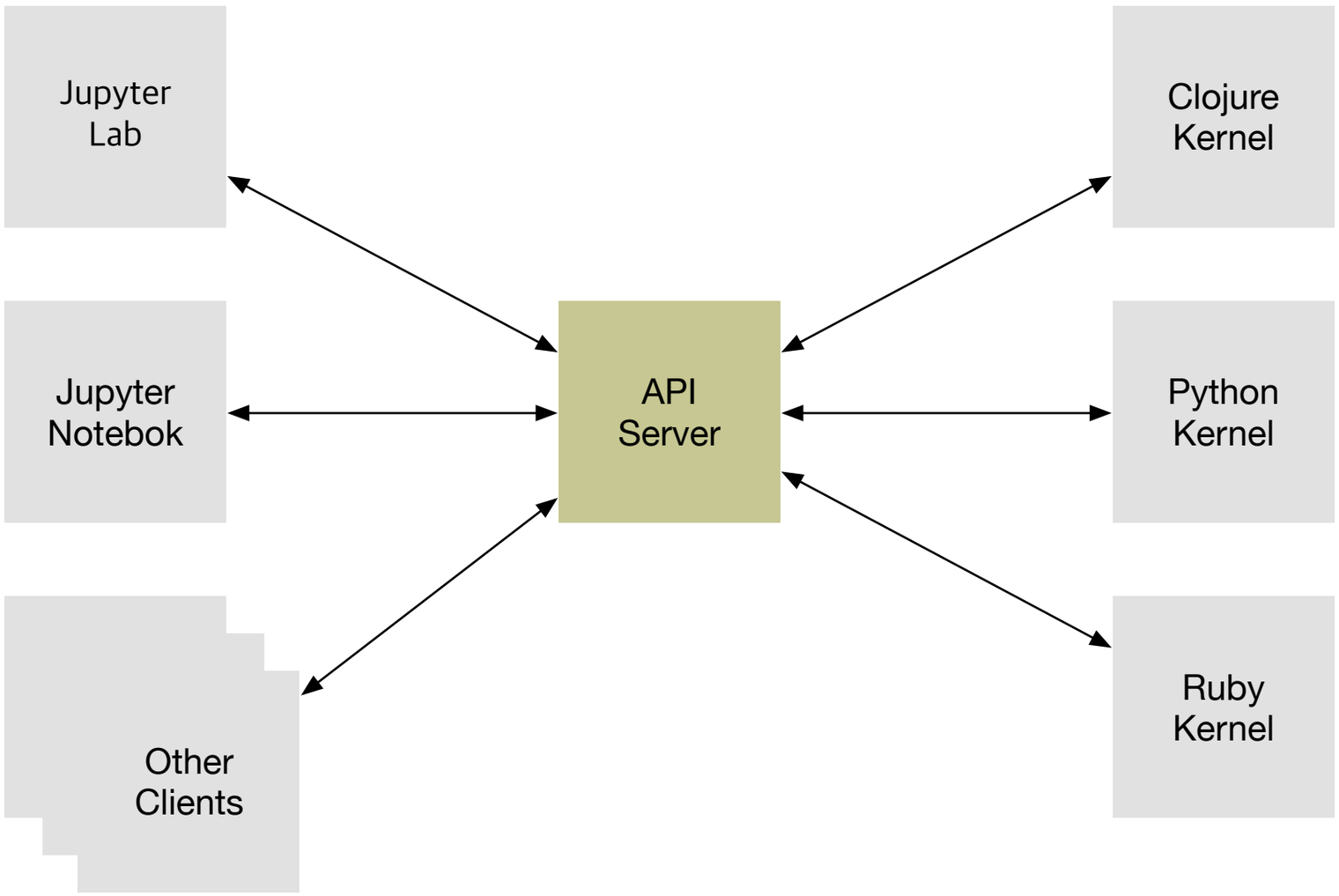
Jupyter Console 5.0.0

정리

```
$ jupyter notebook
```



범용적 REPL 프레임워크



0.0.1

0.11

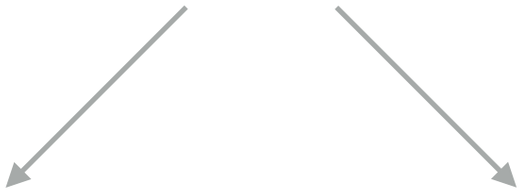
0.12

0.13

1.0

2.0

3.0



IPython 4.0

Notebook 4.0

IPython 5.0

IPython 5.0

Jupyter 1.0.0

Jupyter Notebook 4.2.2

Jupyter Console 5.0.0

Jupyter:

범용적 REPL 프레임워크

범용성

매체성

Jupyter

범용성

매체성

Jupyter: Literate Computing

범용성

매체성

Jupyter: Literate Computing

인터렉티브 노트 저작 도구

범용성

매체성

Write the Docs
Seoul Meetup #1

- 프로그래밍 책을 쓰기 위한 환경 구성과 책을 잘 쓰기 위한 도구
- 자동 조판 워크플로우 실험
- 문서 국제화
- 디지털 미디어 네이티브 문서: Explorable Explanations
- 글을 잘 쓰고 싶은 사람들을 위한 워드 사용법

Write the Docs
Seoul Meetup #1

강남역 메리츠타워 21층 아카마이 코리아

8월 15일(월요일) 오전 10시 ~ 2시

Q&A

SMART STUDY

@nacyo_t

PYCON APAC 2016

Thank you!

SMART STUDY

@nacyo_t

PYCON APAC 2016