

A detailed painting of a scene featuring a window with green shutters. A ginger and white cat is sitting on the windowsill, looking out. The window is framed by a dense arrangement of yellow flowers and green leaves. To the left, a vine with green leaves hangs down. Below the window, there are stone steps and more flowers, including white and pink ones. The overall style is soft and painterly, with warm lighting.

# Python C API

Victor  
Stinner



# My C API long term goals

---



- More **freedom** to change Python **internals**
- Reduce 3<sup>rd</sup> party **maintenance burden** when updating Python (goal: no change)
- Reduce **friction/stress** related to the C API, on the Python side, and on the 3<sup>rd</sup> party side



Red Hat

# My C API long term goals

---



Limited C API by default!  
Stable ABI for all!



**Red Hat**

# Challenges

---



- **Need to change** some 3<sup>rd</sup> party code to reach this long term goal
- Usually **1-10 lines** per Python release in the few **impacted projects**
- **Unknown number** of impacted projects per change



**Red Hat**





Present (3.13)



# Remove private funcs

---



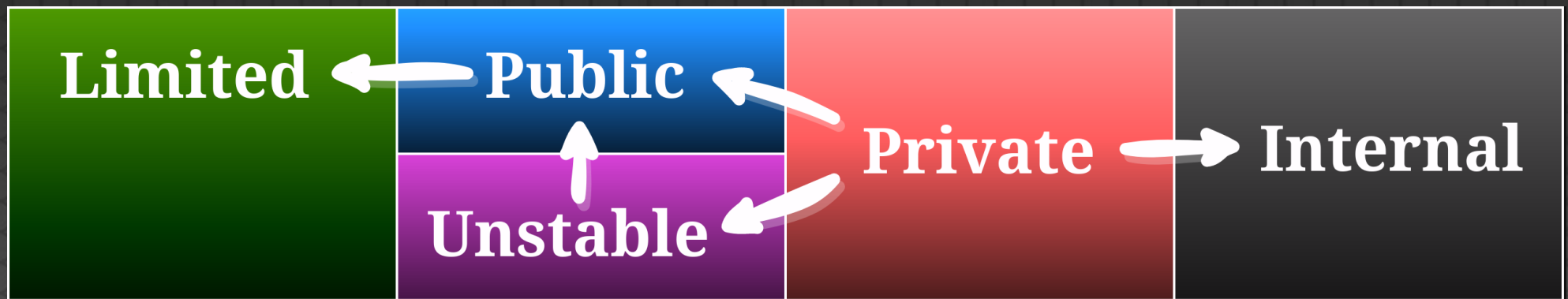
- Removed 300+ **private** functions in Python 3.13 alpha1
- Some removals impacted **5-30 projects**
- As planned, removals causing most trouble have been **reverted** (mostly in alpha2)
- 3.13 beta1: **264** functions removed



Red Hat

# Remove private funcs

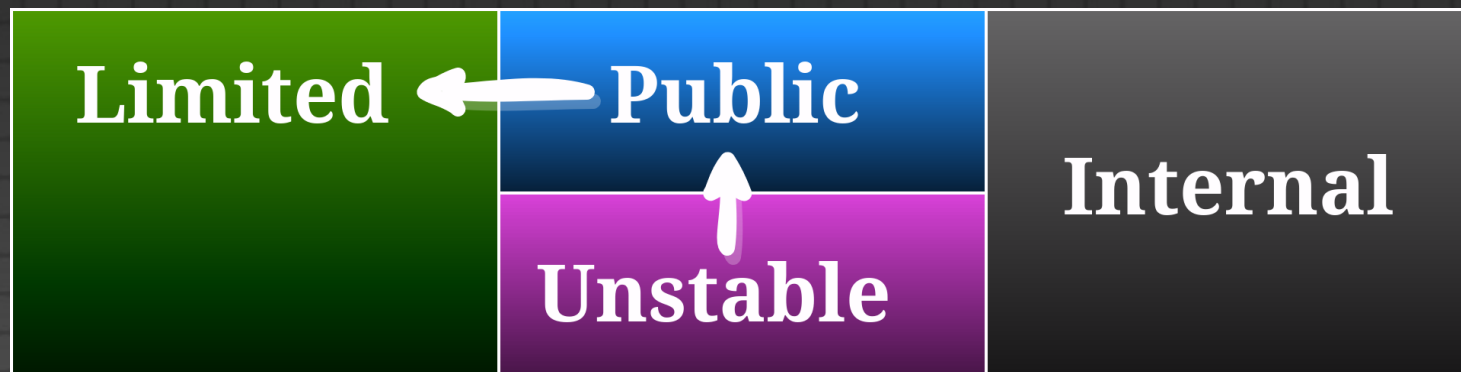
---



Red Hat

# Remove private funcs

---



Red Hat



# Better public functions

---



- Promote private to **public** functions
- Add **backward compatibility warranty**
- Add **documentation** and **tests**
- C API Working Group designs a **better** API
- Enforce error checking
- Avoid inefficient `PyErr_Occurred()` check



Red Hat

# PyDict\_Pop() example

---



- 3.12: **PyObject\*** \_PyDict\_Pop(PyObject \*dict, PyObject \*key, PyObject \*default)  
→ NULL means "not found" **or** error
- 3.13: **int** PyDict\_Pop(PyObject \*op, PyObject \*key, **PyObject \*\*result**)
  - Return **-1** on **error** and set an exception
  - Return **0** if **not found** (NULL result)
  - Return **1** if **found** (non-NULL result)



Red Hat



# Argument Clinic Limited API

---



- Use limited C API if code uses **#define Py\_LIMITED\_API** ...
- Only use **public** functions of the limited C API
- **Inline code** to avoid calling private/internal functions



Red Hat

# AC: fcntl.fcntl() clinic input

---



```
fd: files  
cmd as code: int  
arg: object(c_default='NULL') = 0  
/
```



Red Hat



# AC:fcntl.fcntl() code in 3.12

---



```
if (!_PyArg_CheckPositional(  
    "fcntl", nargs, 2, 3)) ...  
if (!_PyLong_FileDescriptor_Converter(  
    args[0], &fd)) ...  
code = _PyLong_AsInt(args[1]);
```



Red Hat

# AC:fcntl.fcntl() code in 3.13

---



```
if (nargs < 2) { PyErr_Format(...)... }
if (nargs > 3) { PyErr_Format(...)... }
fd = PyObject_AsFileDescriptor(args[0]);
if (fd < 0) ...
code = PyLong_AsInt(args[1]);
```



Red Hat



# Build stdlib ext with limited

---



- **Build** 16 stdlib extensions with the **limited C API**
- errno, md5, resource, \_uuid, ...
- **Test** our own **limited C API**
- PEP 737 "Unify type formatting" adds **%T** and **%N** formats for error messages



Red Hat

# And Python <= 3.12 ?

---



**pythoncapi-compat** provides new Python 3.13 functions on Python 3.12 and older

- `PyDict_GetItemRef()`
- `PyLong_AsInt()`
- `PyUnicode_EqualToUTF8()`
- ...



**Red Hat**



# Stable ABI fix

---



- Py\_None no longer reads the private symbol: &\_Py\_NoneStruct
- Limited C API 3.13 calls the public function: **Py**\_GetConstantBorrowed(Py\_CONSTANT\_NONE)
- New Py\_GetConstant(id) API



Red Hat



Future  
(3.14)



# Limited C API & stakeholders



- Cython: **opt-in** build mode
- PyO3 (Rust): **opt-in** build mode; consider making it the default!
- pybind11 (C++11): **not** supported
- nanobind (C++11): 100% supported (is it the **default**?)
- cffi: **not** supported



Red Hat

# abi3audit

---



- Test if a binary uses the stable ABI or not
- <https://github.com/trailofbits/abi3audit>



**Red Hat**



# TODO

---



- Add API for **PyFrameObject**  
Cython, greenlet, eventlet, gevent, ...
- Add API for **PyThreadState**  
Cython, ...
- Design a migration path away from **PyObject** members direct access



Red Hat