

# Architecture

Architecture

Anonymous · 10/10/2026

---

## Architecture

`phpy` is only ever imported by `GenericDispatcher`, `GenericExplorer`, and the internal `_phpy_conversions` module they both share for turning a live PHP object into a plain Python one — never by the tests, a specific library's own bridge, or the application consuming it. Building one specific library's bridge means subclassing `GenericDispatcher`/`GenericExplorer` with its `bootstrap_class`, and never touching `phpy` directly:

```
class GenericDispatcher:
    _AUTOLOAD_PATH_ENV = 'BACKBONE_DISPATCHER_AUTOLOAD'

    def __init__(
        self,
        bootstrap_class: str,
        bootstrap_method: str = 'boot',
        bootstrap_args: tuple = (),
        exception_registry: ExceptionRegistry | None = None,
        autoload_path: str | None = None,
    ) -> None: ...

    def dispatch(self, operation_id: str, **params) -> OperationResult: ...
```

`ExceptionRegistry` (the PHP-class-name → Python-exception mapping, via `raise_for(problem, metadata=None)`), and the `OperationResult`/`Problem`/`SafeThrowable`/`ExecutionMetadata` dataclasses themselves, are all separate, independent components that never touch `phpy` — plain, immutable data holders that can be built, tested and reasoned about with no PHP interpreter involved at all. `_phpy_conversions` is the one place that bridges the two worlds, turning a live PHP object into one of those.

---

Last updated on 10/10/2026