Strix

Why Standards Matter

How We Unified our Setups to Save Time and Foster Growth

André Varelmann
Head of Solution Architecture
andre.varelmann@strix.net

We are Strix

We are pushing boundaries in digital commerce

3

NEW

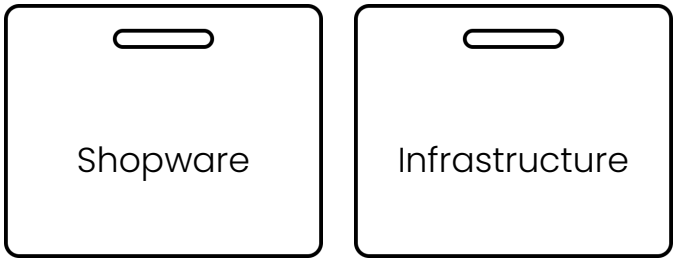
Project

NEW

Project

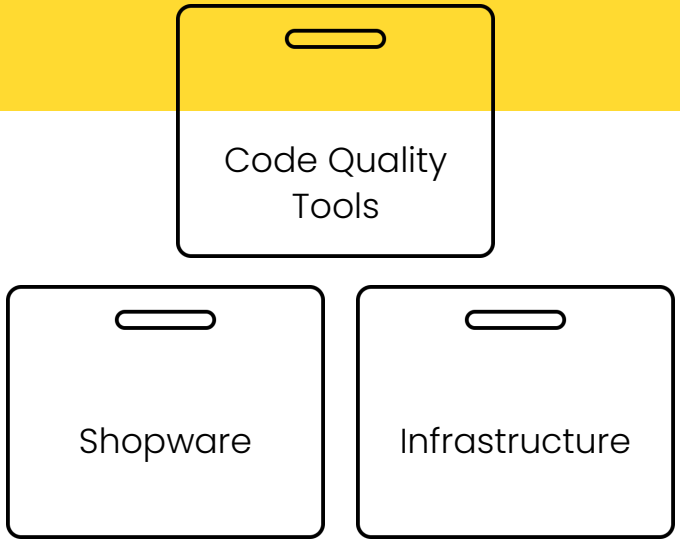
NEW

Project



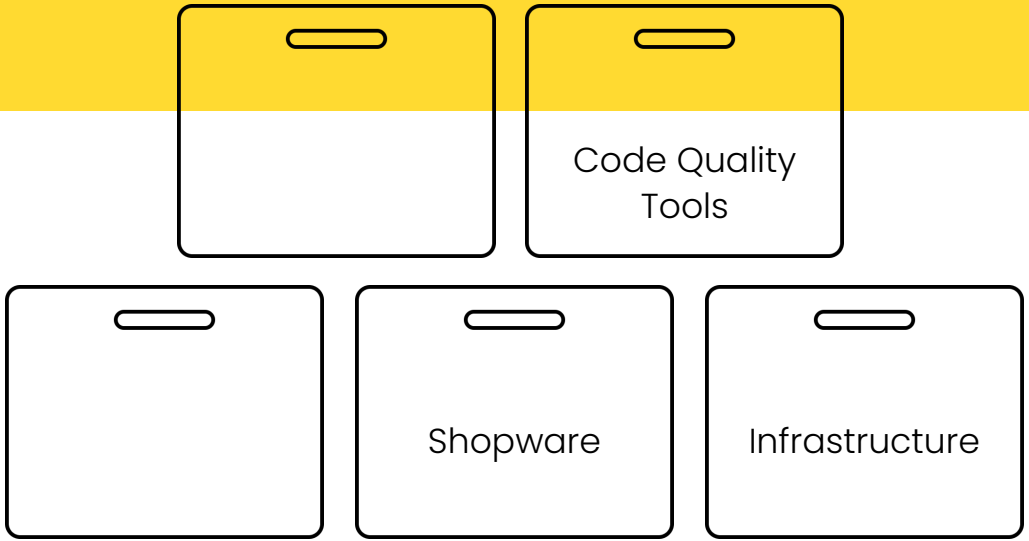
NEW

Project



NEW

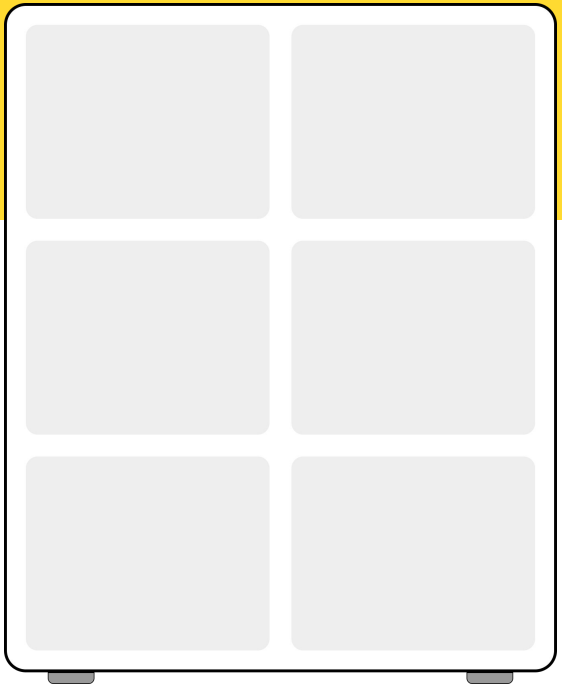
Project



Let’s clean up

Introducing Standards

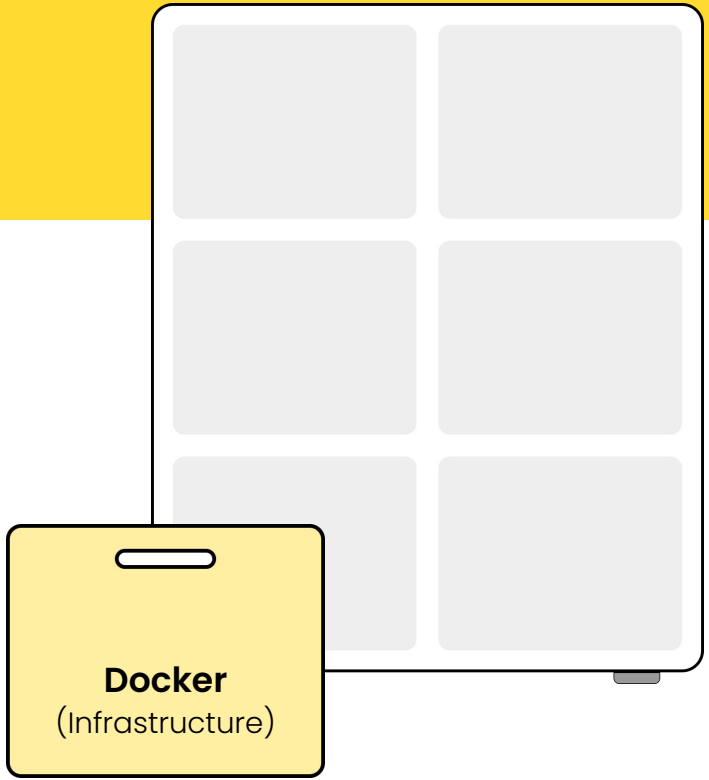
- One foundation for all projects
- A base set of tools and helpers
- Easy to use in daily project life
- Maintainable and upgradable



Docker

Dockware as the Standard

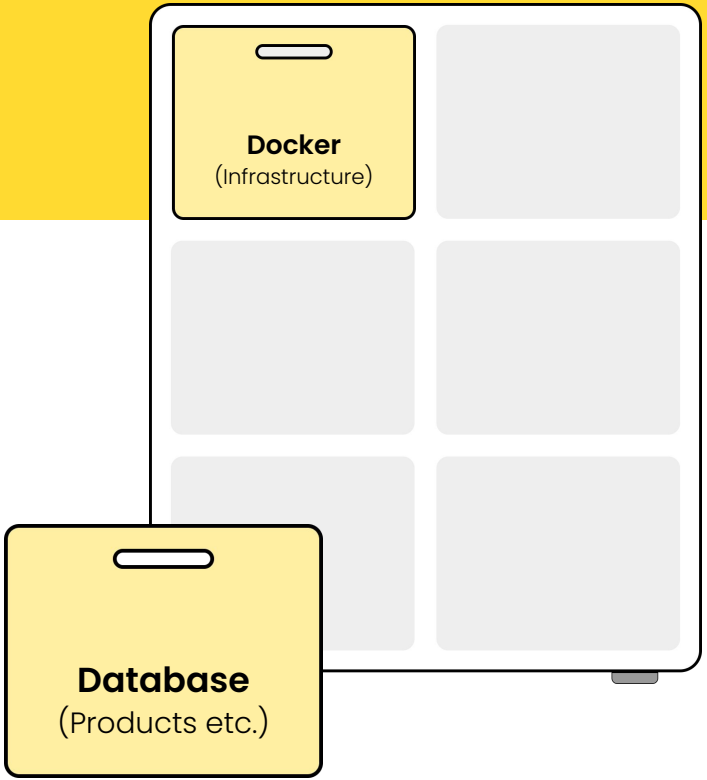
- All projects run on dockware 💛
- Quick and consistent setup
- Added preconfigured environment files
- Also used in github workflows for integration tests



Database

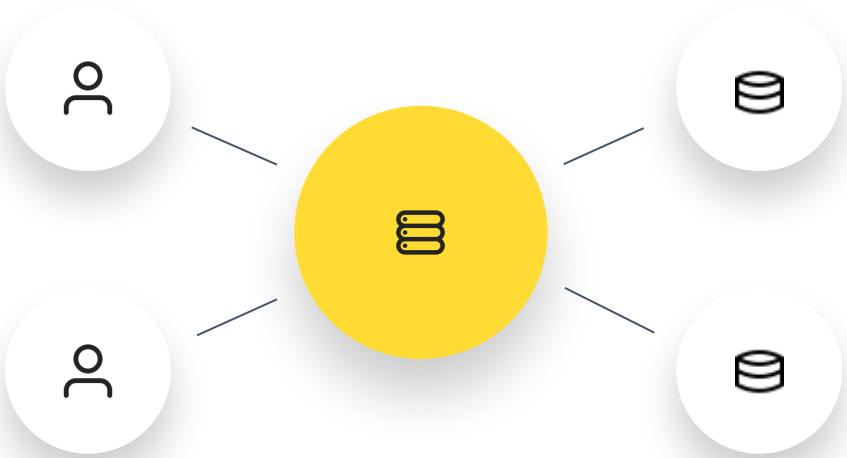
Using the Client database for local development

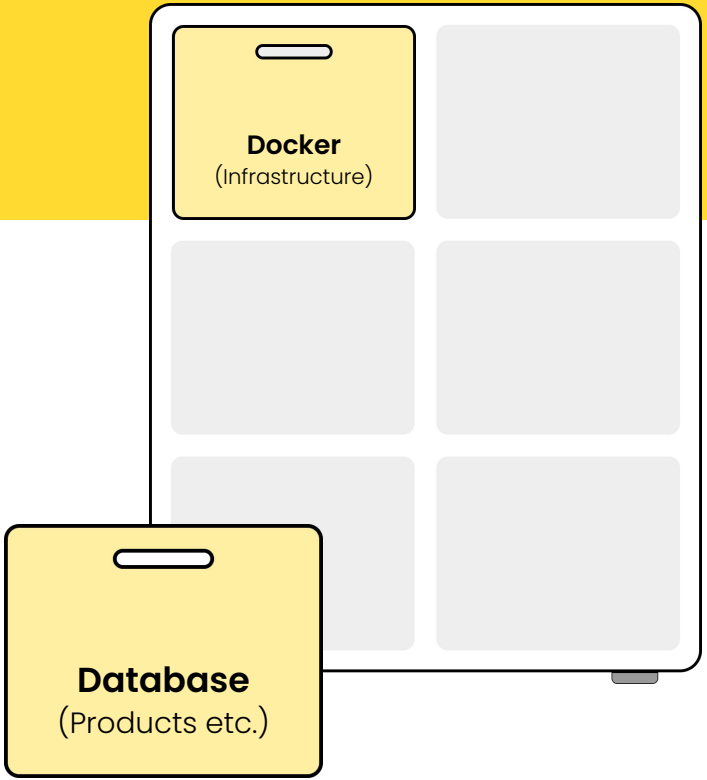
- Database dumps are created daily or on demand
- Data anonymized automatically by *gdpr-dump*
- Proxy server for database downloads



Database

Proxy Server

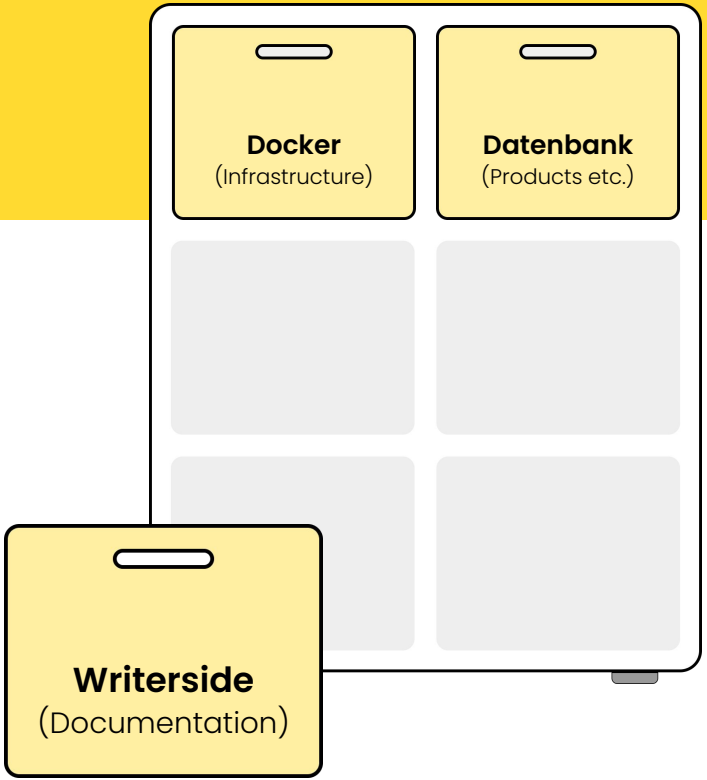




Technical Documentation

Writerside

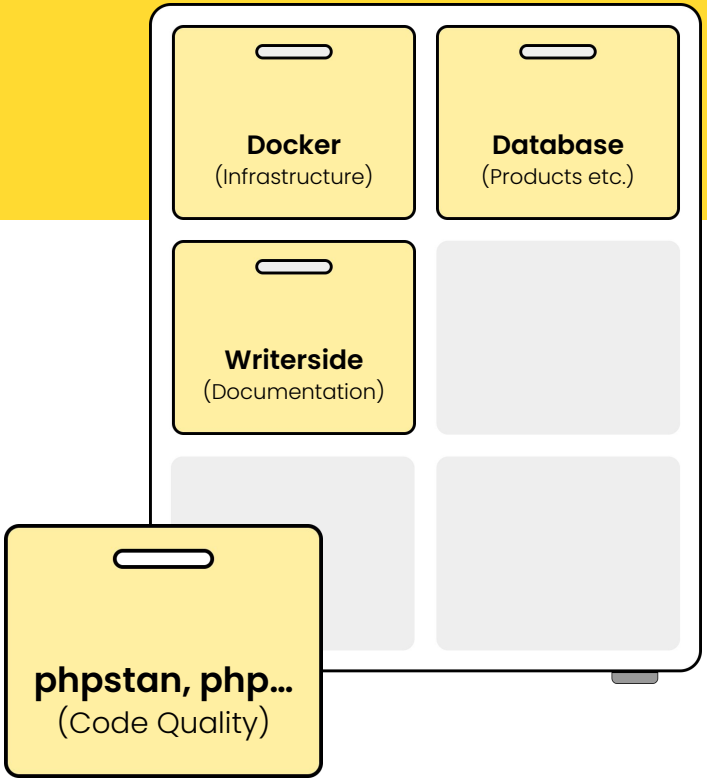
- Unified technical documentation in one tool
- Documentation templates can be used
- Automatic deployment of documentation
- Encourage knowledge exchange



Quality Assurance

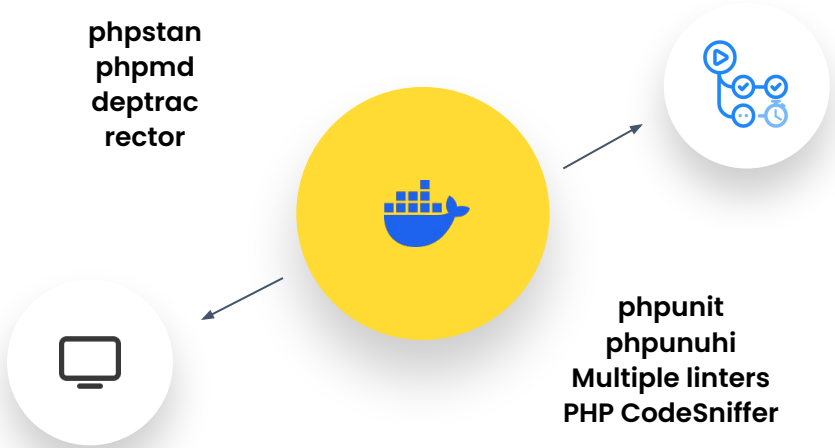
Containerized QA

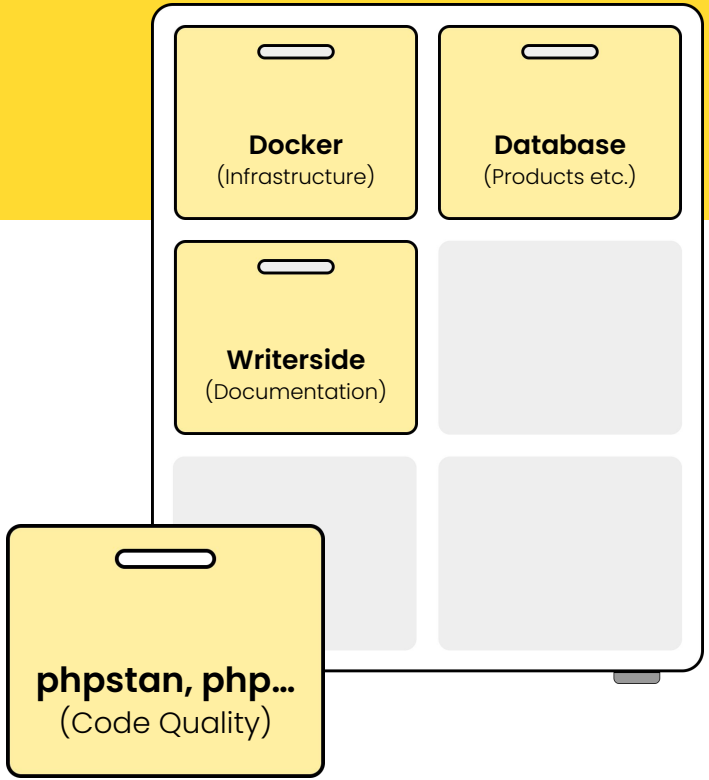
- Decided on a base set of QA tools
- Defined standard configuration
- Running inside docker container



Quality Assurance

Containerized QA

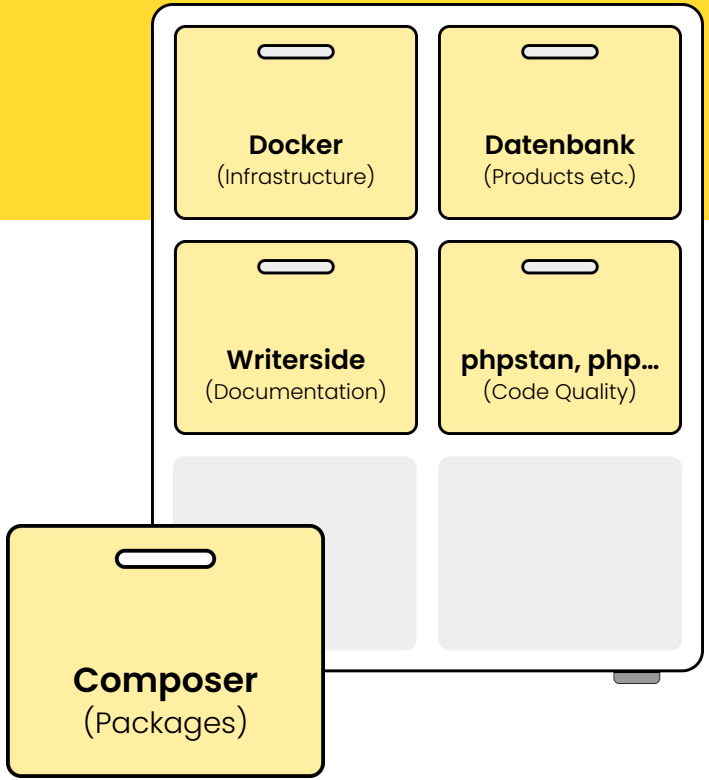




Package Management

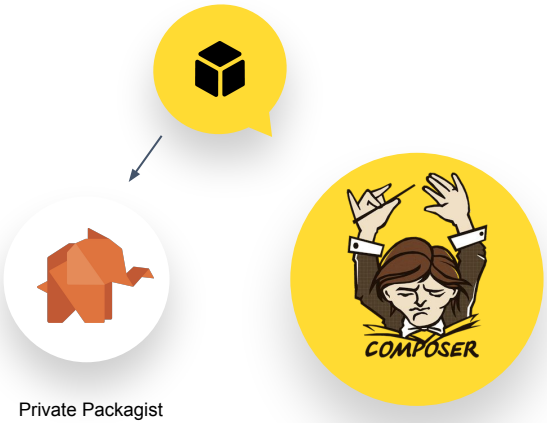
Component based development

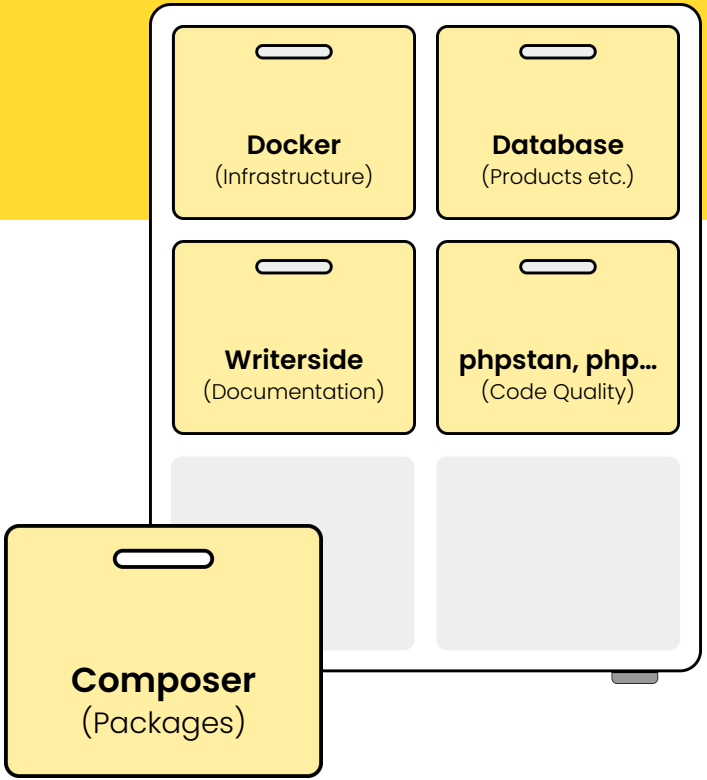
- Introduced private packagist
- Development helpers now come pre installed
- Automatic configuration with *symfony/flex*
- **Requires mindset shift in development**



symfony/flex

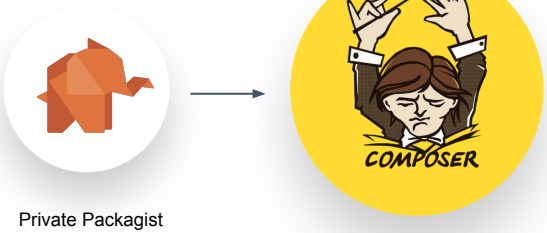
How it works





symfony/flex

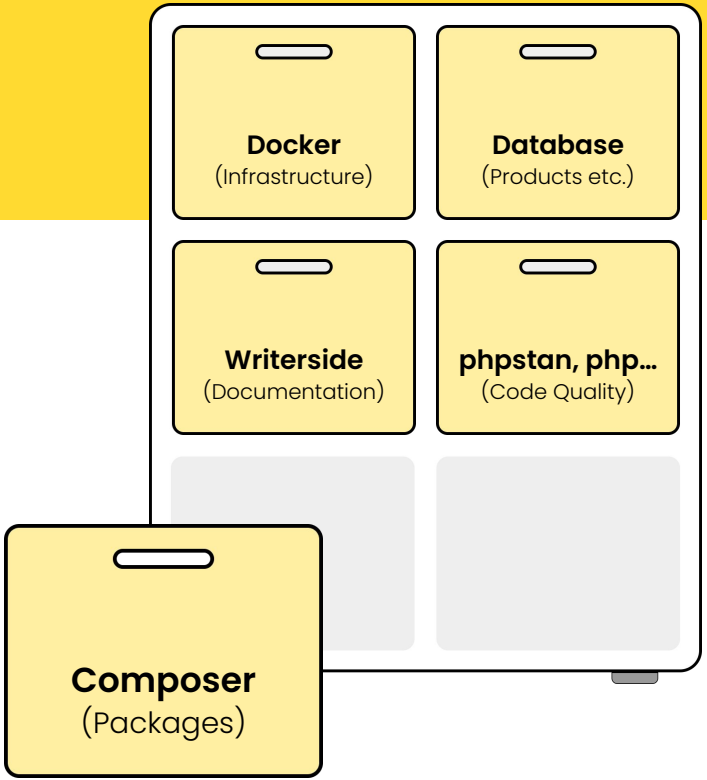
How it works



Private Packagist

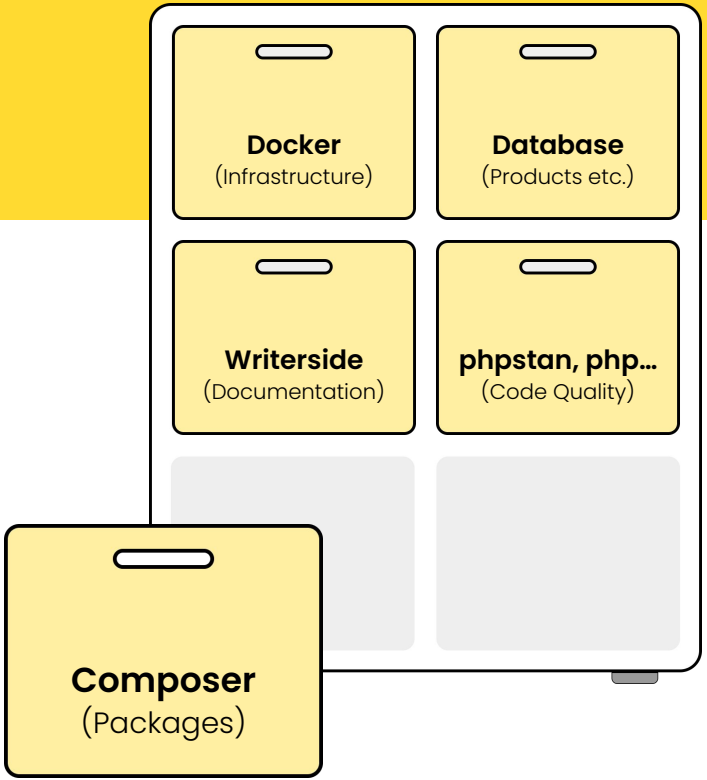
Symfony

Recipes



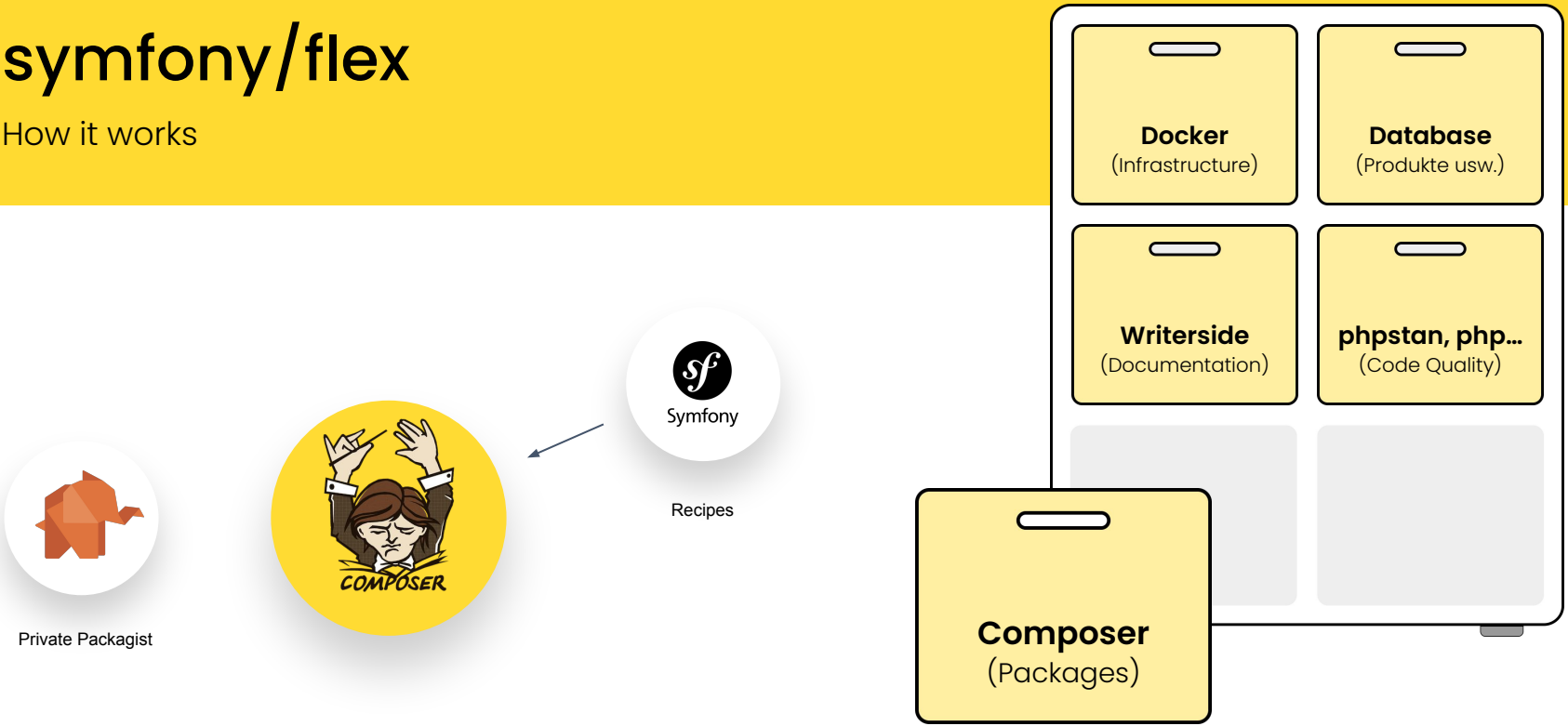
symfony/flex

How it works



symfony/flex

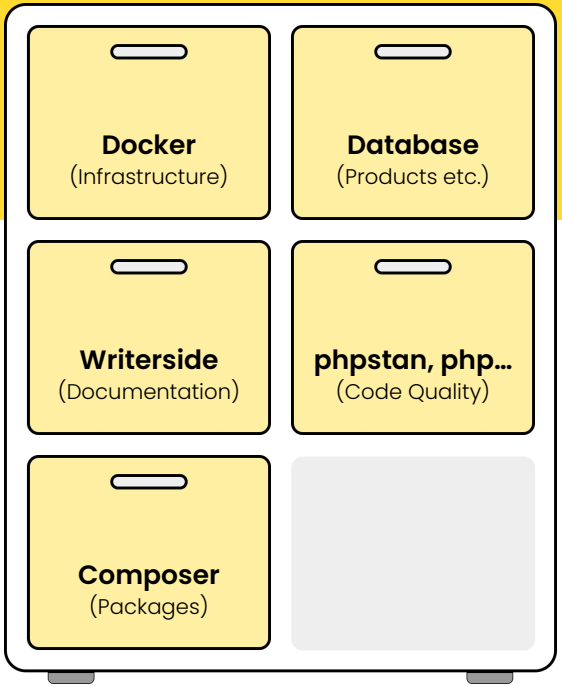
How it works



Shopware Project Skeleton

That's it, right?

- Adding some scripts to start the setup
- Skeleton Template combines all standards
- New projects are created from github template
- Existing projects need to be migrated once

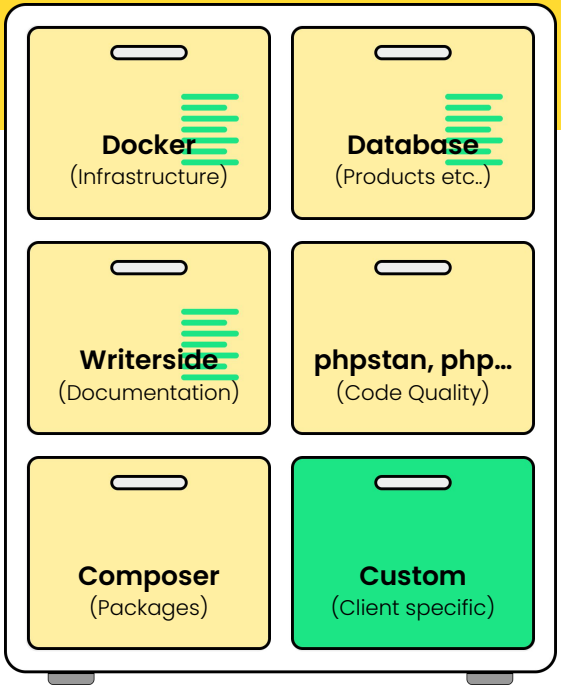


Projects evolve individually

Projects evolve individually

We need an upgrade path!

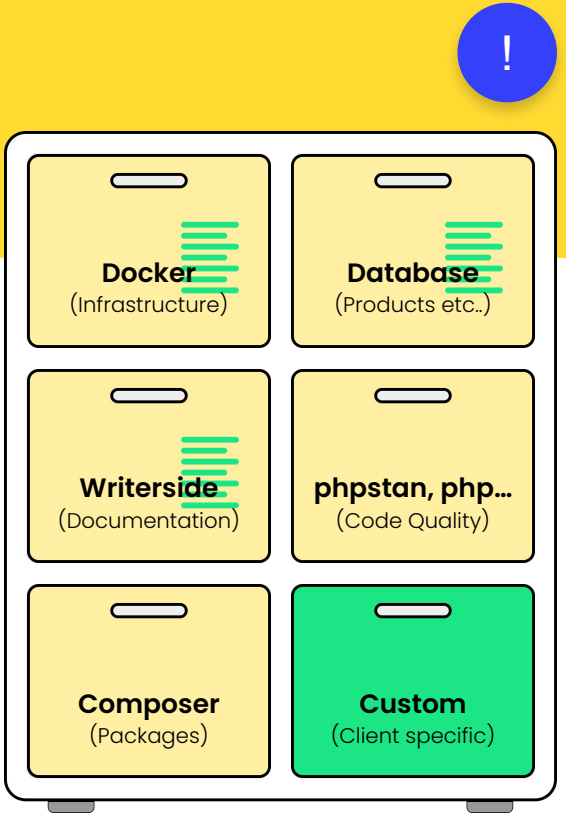
- Custom Config & Code is added
- Setups are adjusted to project needs
- Currently: Manual Upgrade Path
- **How to prevent falling into old patterns?**



Modularity

Configuration based on flex recipes

- Split up the skeleton into components
- Based on composer meta packages
- Created flex recipes for all components
- Updates via *composer recipes:update*



```
{
  "name": "strix/skeleton-dev-ops-shopware",
  "type": "metapackage",
  "require": {
    "php": ">=8.1",
    "phpmd/phpmd": "^2.15",
    "phpstan/phpstan": "^2.1.2",
    "symfony/dependency-injection": "^7.0.10",
    "phpstan/phpstan-doctrine": "^2.0.1",
    "phpstan/extension-installer": "^1.4.3",
    "squizlabs/php_codesniffer": "^3.8",
    "slevomat/coding-standard": "^8.14.1",
    "qossmic/deptrac-shim": "^1.0.2",
    "rector/rector": "^2.0.7",
    "frosh/shopware-rector": "^0.5.1",
    "boxblinkracer/phpunuhi": "^1.21"
  },
  "conflict": {
    "shopware/core": "*"
  }
}
```

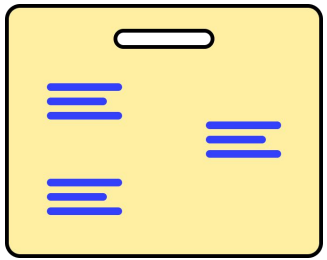
```
{
  "copy-from-recipe": {
    "github/": ".github/",
    "src/": "src/"
  },
  "docker-compose": {
    "docker-compose.yml": {
      "services": [
        "qa:",
        "  build:",
        "    context: dev-ops/docker/qa",
        "    args:",
        "      - PHP_VERSION=${PHP_VERSION:-8.4}",
        "  tty: true",
        "  working_dir: /app/",
        "  volumes:",
        "    - './:/app/'"
      ]
    }
  },
  "composer-commands": {
    "auto-config-validate": "./tools/auto-config-validate",
    "deptrac": "./tools/deptrac",
    "phpcs": "./tools/phpcs",
    "qa": ["@deptrac", "@phpcs", "@phpmd", "@phpstan", "@phpunuhi", "@rector"],
    "qa-integration": ["@qa", "@auto-config-validate", "@phpunit", "@twiglint"]
  }
}
```

Applying Recipe Updates

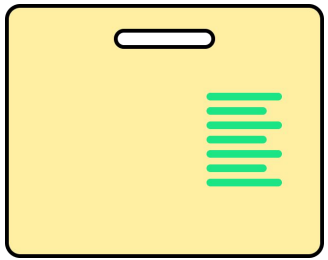
Comparing versions

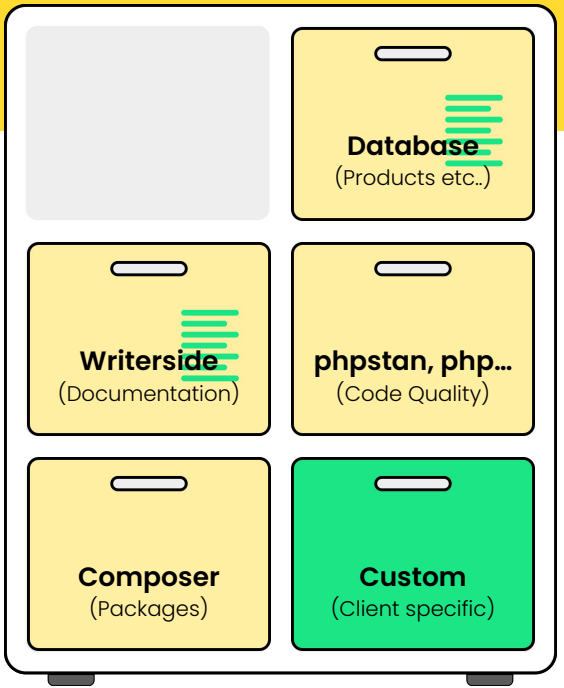
composer recipes:update

Skeleton Update



Version Client

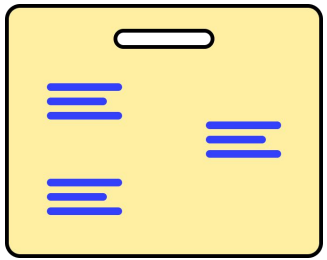




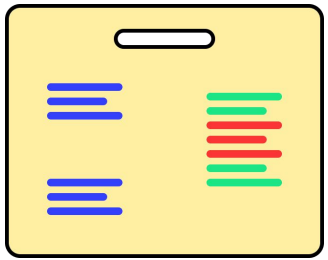
Applying recipe updates

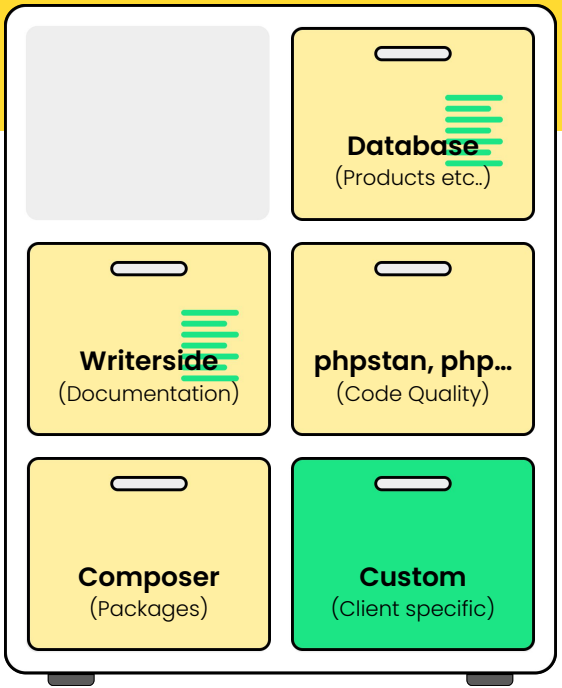
Merging changes

Skeleton Update



Version Client



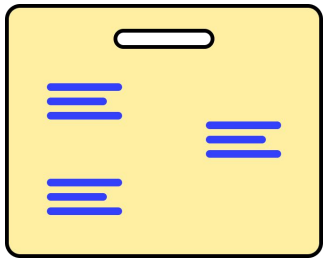


Applying recipe updates

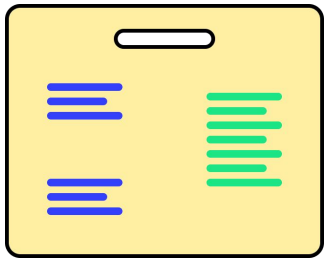
Update applied

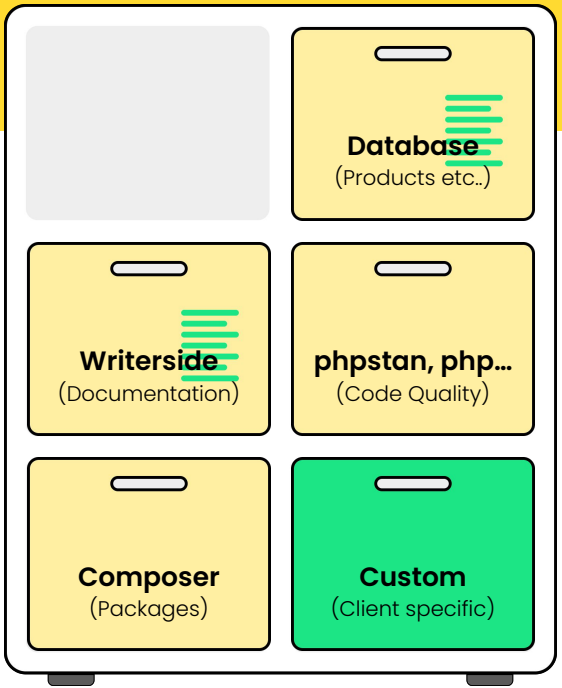
git commit

Skeleton Update



Version Client

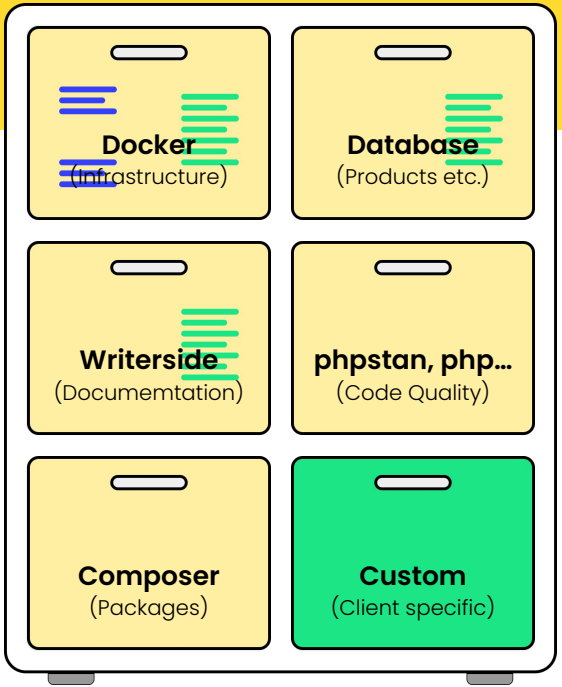




Wrap Up

What did we achieve?

- Stable foundation for new and existing projects
- Switching between projects made easy
- Component based and upgradable
- **Allows developers to focus on what really matters**



Strix

Let’s talk

André Varelmann
Head of Solution Architecture
andre.varelmann@strix.net