

Soyons fou ! Et si Drupal faisait moins de chose au runtime ?

Par Benjamin Rambaud
aka beram



Soyons fou ! Et si Drupal faisait moins de chose au runtime ?

Par Benjamin Rambaud
aka beram



Disclaimer



Hello! I'm beram 🖐️

🐷 Benjamin Rambaud
🐘 PHP Engineer at [@ekino_France](#)

- Twitter: [@rambaud_b](#)
- Gitlab: [beram](#)
- Github: [brambaud](#)
- Drupal.org: <https://drupal.org/u/beram>
- Blog: <https://brambaud.github.io/>





Let's dream!



Let's dream!

Simplicity

Predictability



Performance

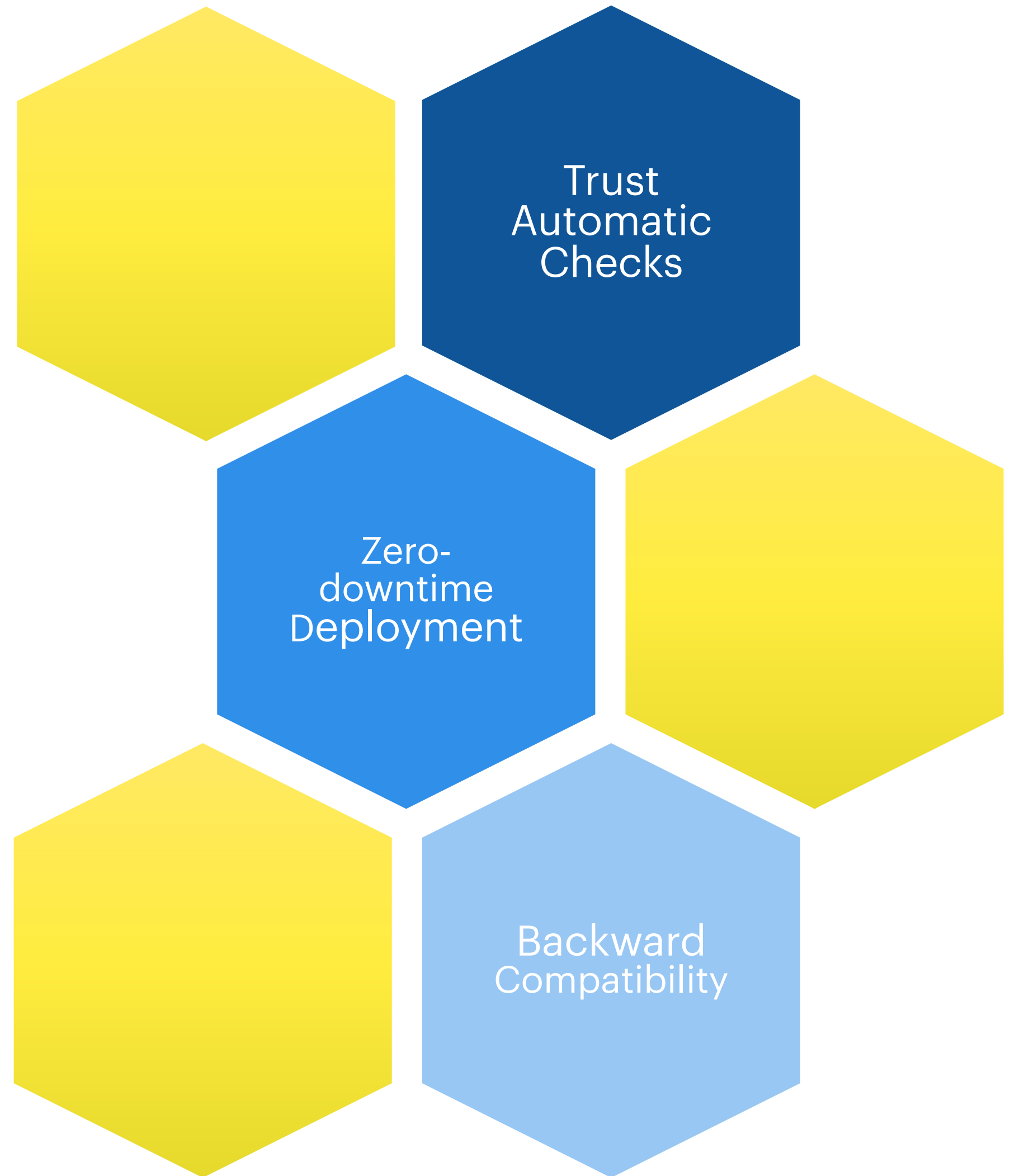
Serenely focus on what matters





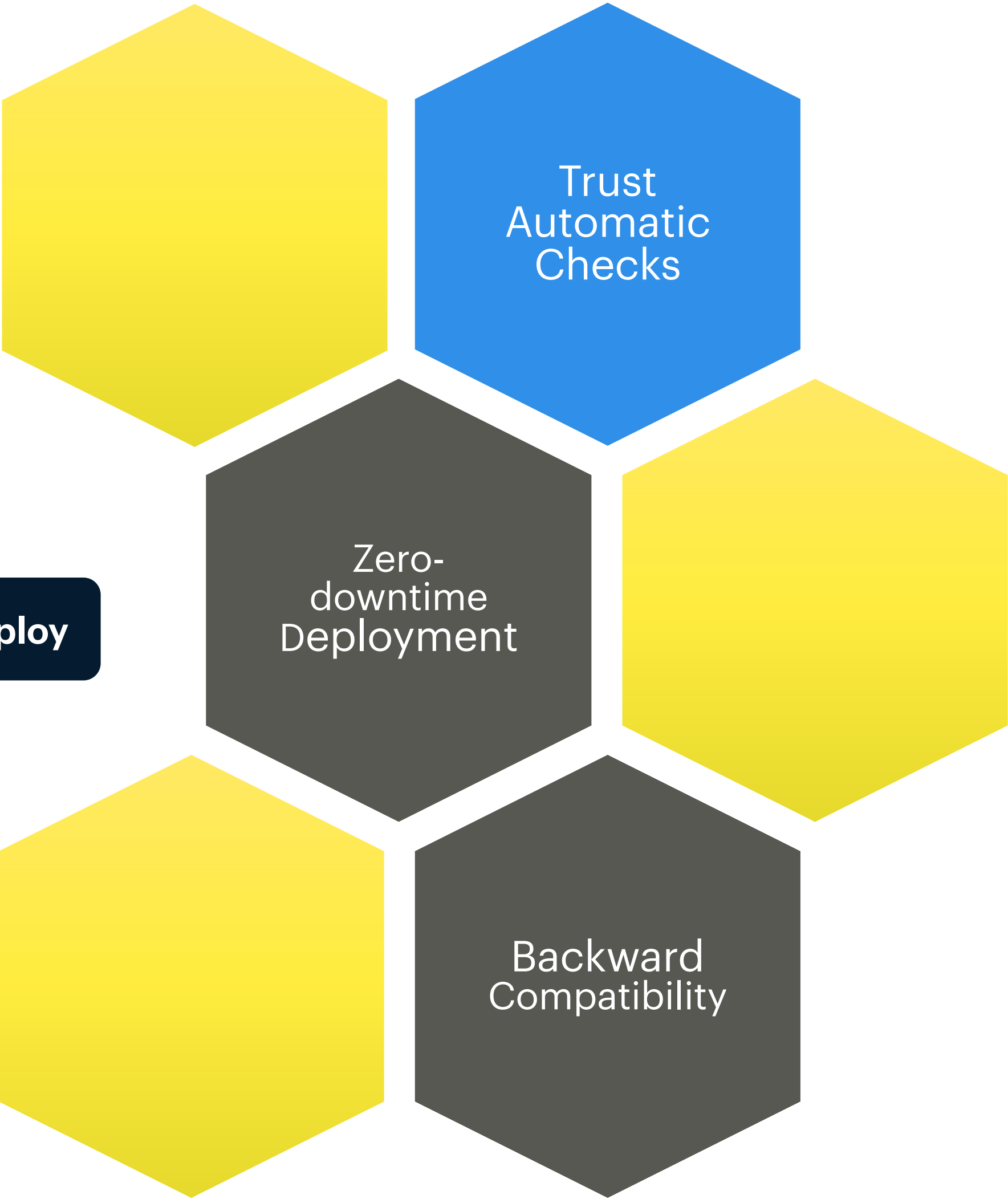
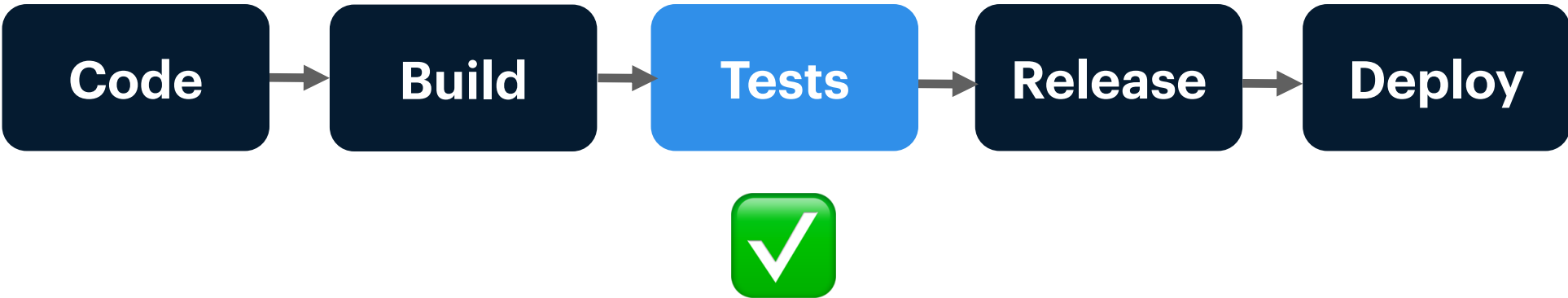
Let's dream!

**Deploy
as many times
needed
per day!**



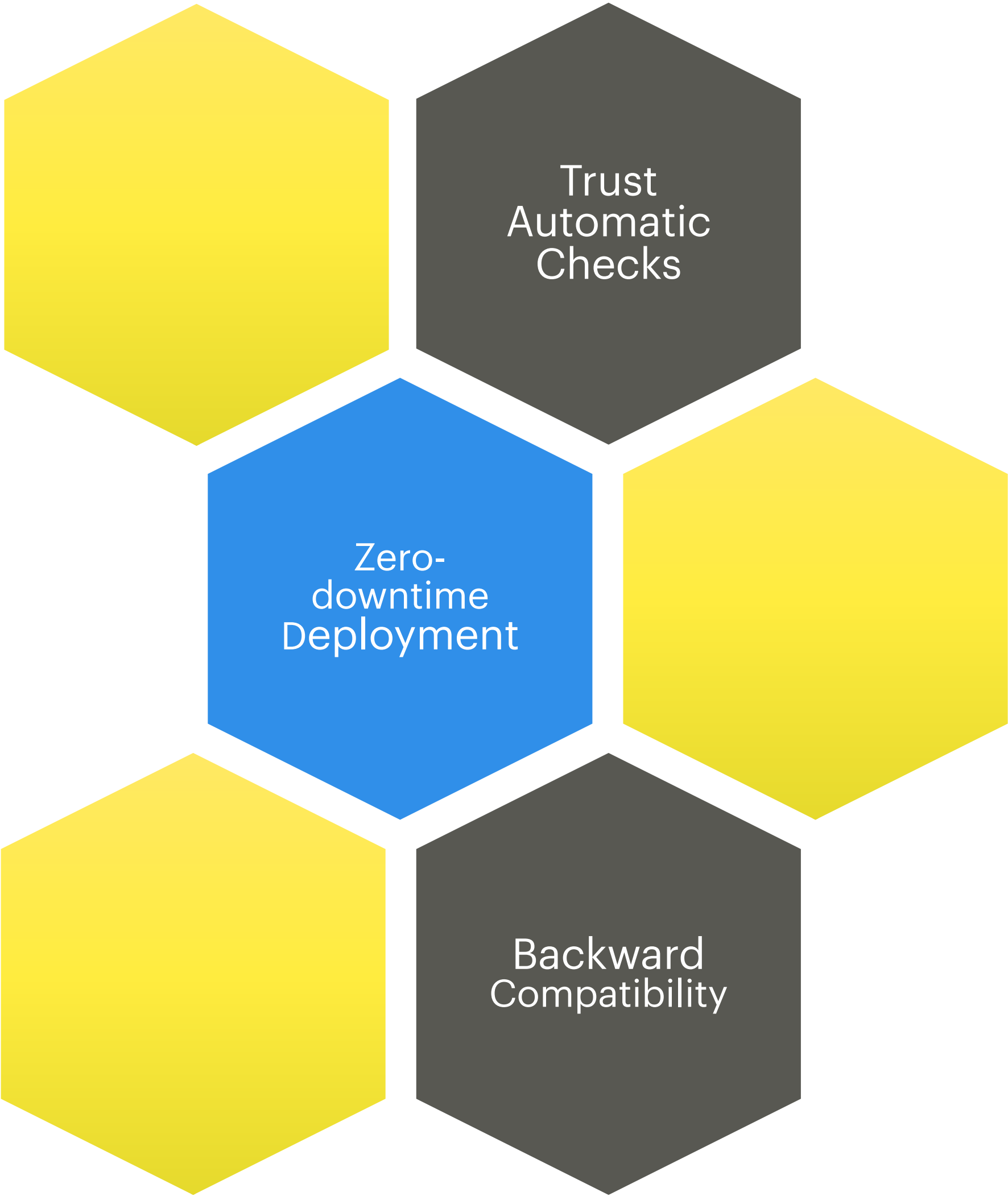


Let's dream!



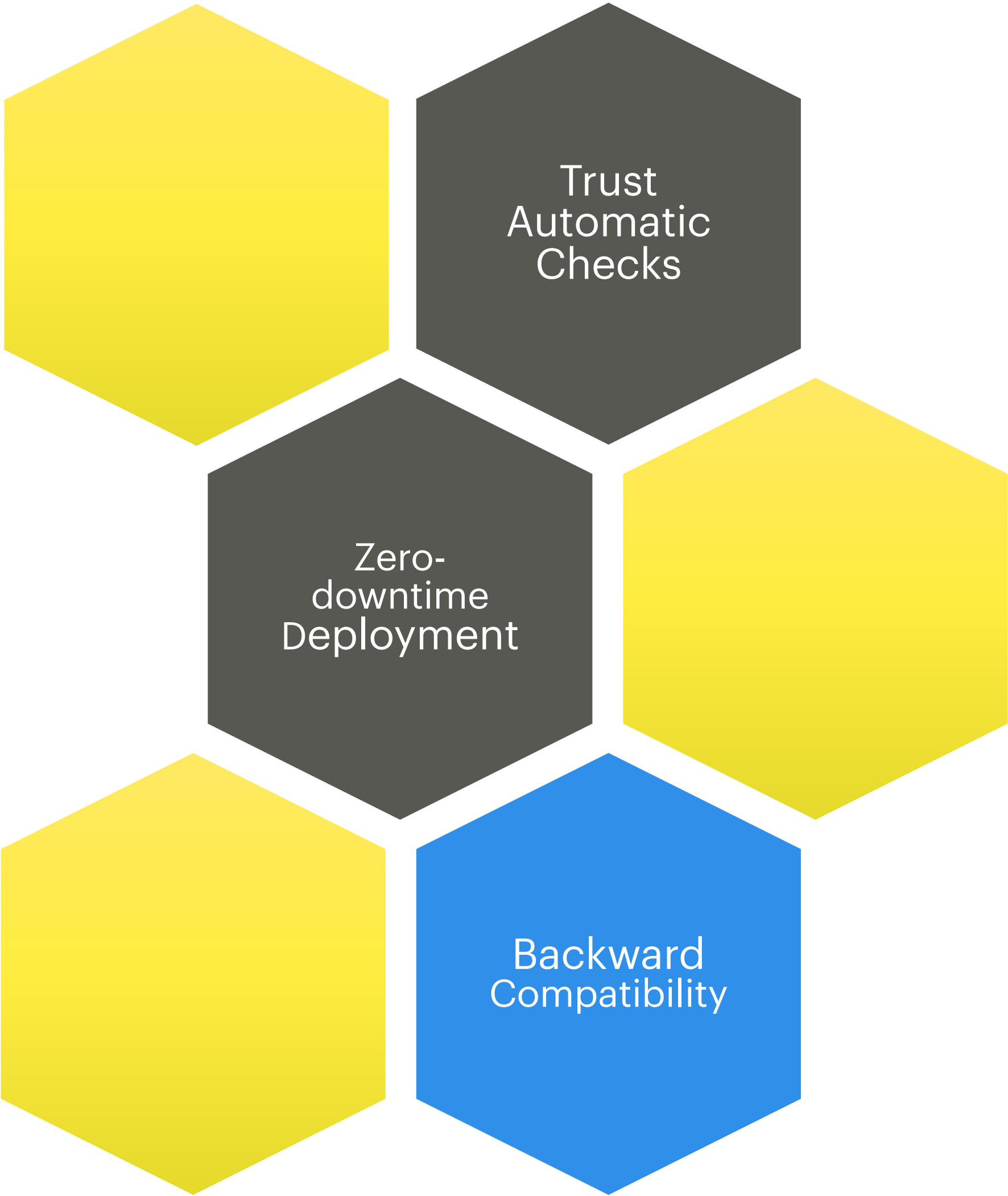
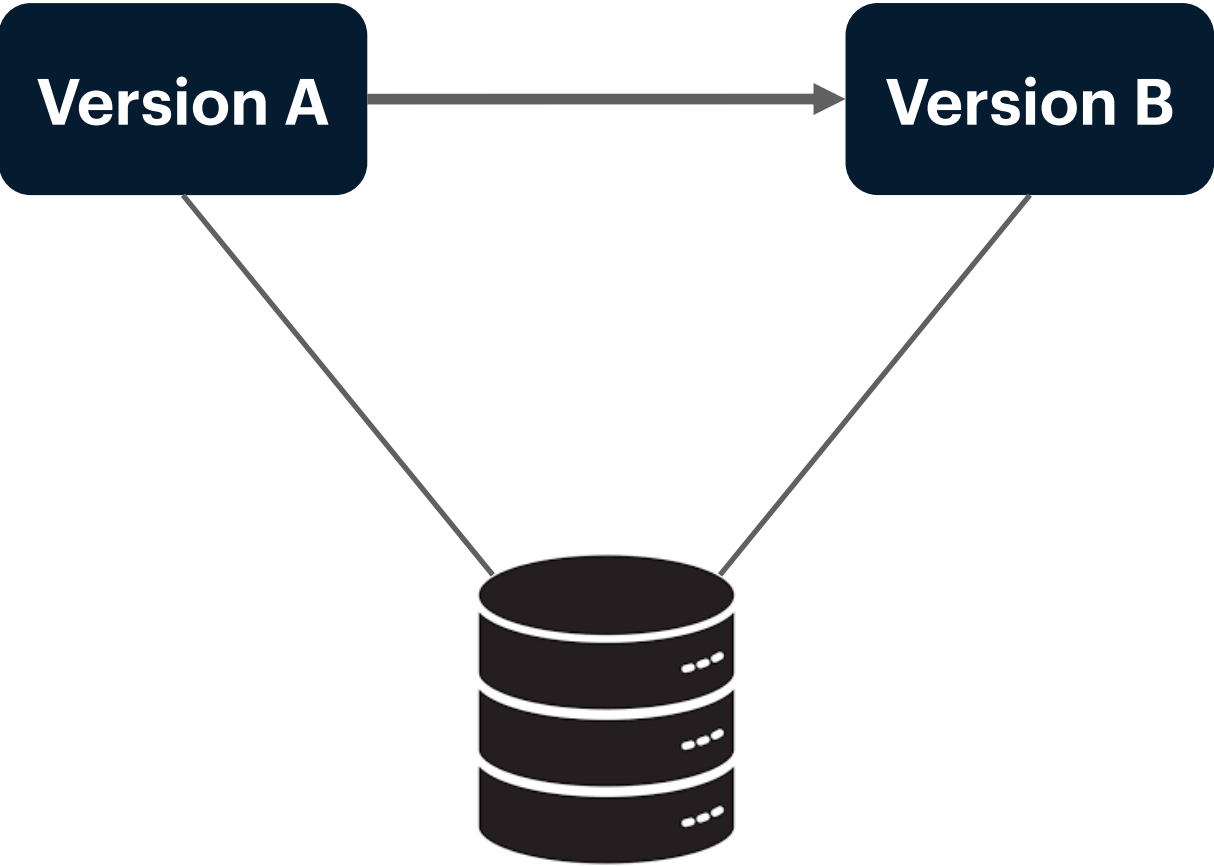


Let's dream!





Let's dream!





Let's dream!

Focus on one aspect!





Let's dream!
Focus on one aspect!

A compilation phase

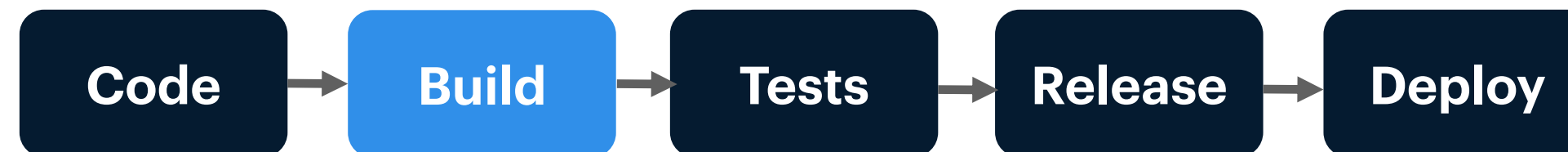
Not this one 🥁



It's only for the joke 😂 Not an ad 🙄



Let's dream!
Focus on one aspect!



A compilation phase

Not this one 🥁



It's only for the joke 😂 Not an ad 🙄



**What to compile?
What could it bring?**

« In production servers,
PHP files should never change,
unless a new application version is
deployed »

<https://symfony.com/doc/current/performance.html>

What to compile?



**Everything
that does not
need to change**



What could it bring to compile... Container

Performance
and
simplicity



+ OPcache

```
<?php // getFooService.php

namespace ContainerNV6rTbr;

class getFooService extends App_KernelProdContainer {
    public static function do($container) {
        return $container->privates['App\\Foo'] = new \\App\\Foo(
            $container->privates['bar'] ?? $container->load('getBarService')
        );
    }
}
```

What could it bring to compile... Container

Predictability

Decrease
downtime



TODO List

~~Compile the container~~
So
Many
Things

What could it bring to compile... Container

Reuse Symfony Component directly

symfony/dependency-injection



What could it bring to compile... Container



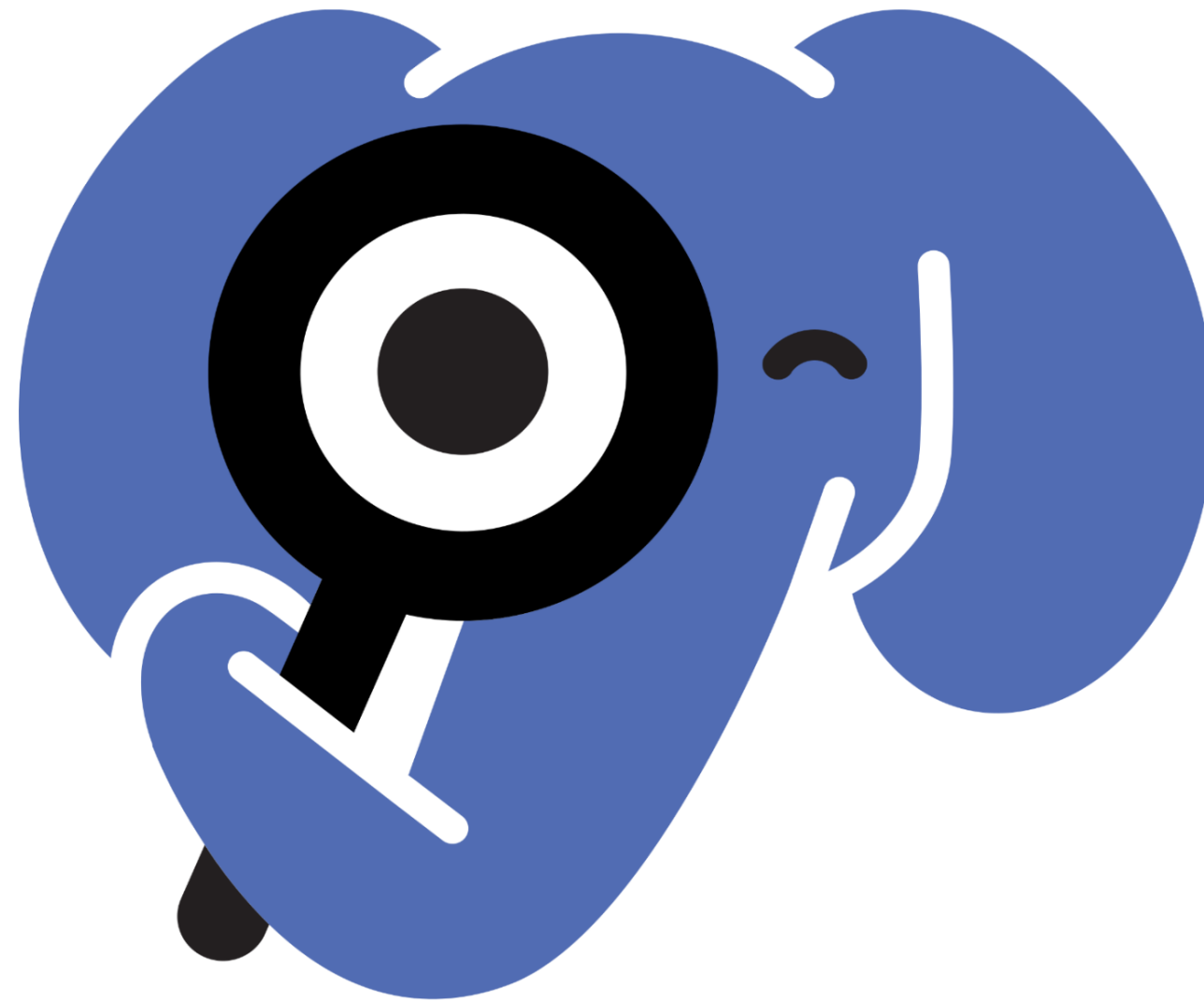
TODO List

~~Compile the container~~
So
Many
Things

What could it bring to compile...

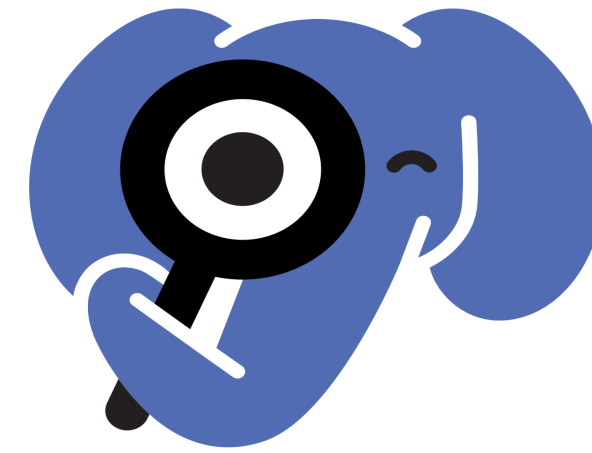
Container

PHPStan

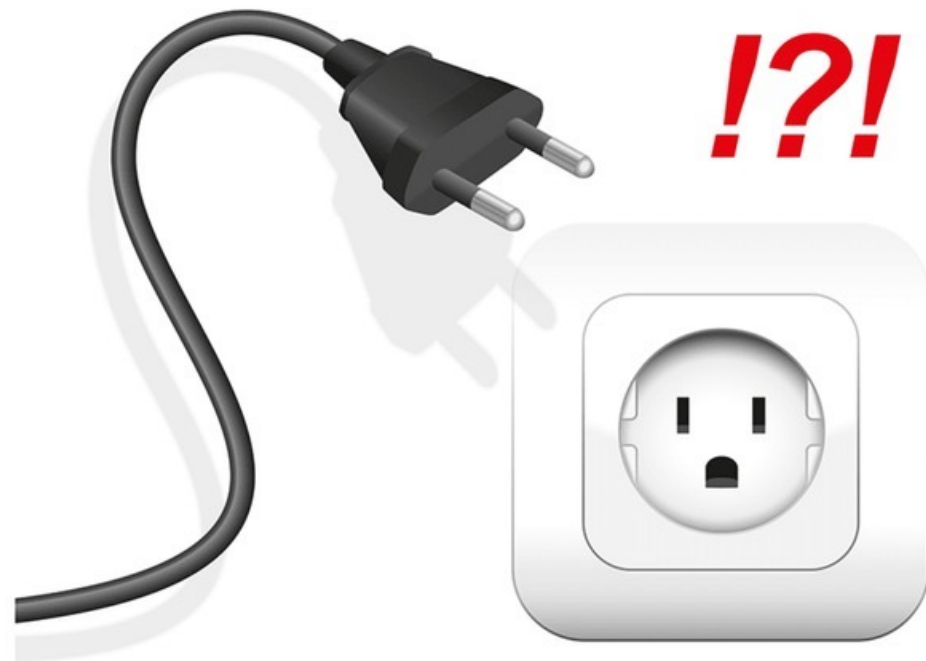


What could it bring to compile... Container

PHPStan



Static === Don't execute the program
Drupal does everything at runtime



Less duplication



What could it bring to compile...
Container



Lots of possibilities!

What could it bring to compile...

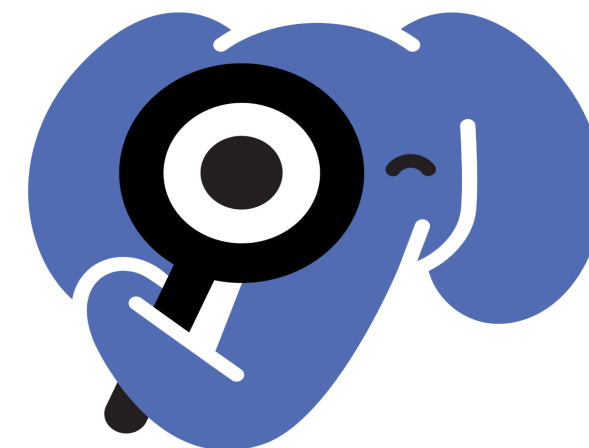
Router

Performance
and
simplicity

Decrease
downtime

Predictability

PHPStan



What could it bring to compile... Entity Definition, storage etc..

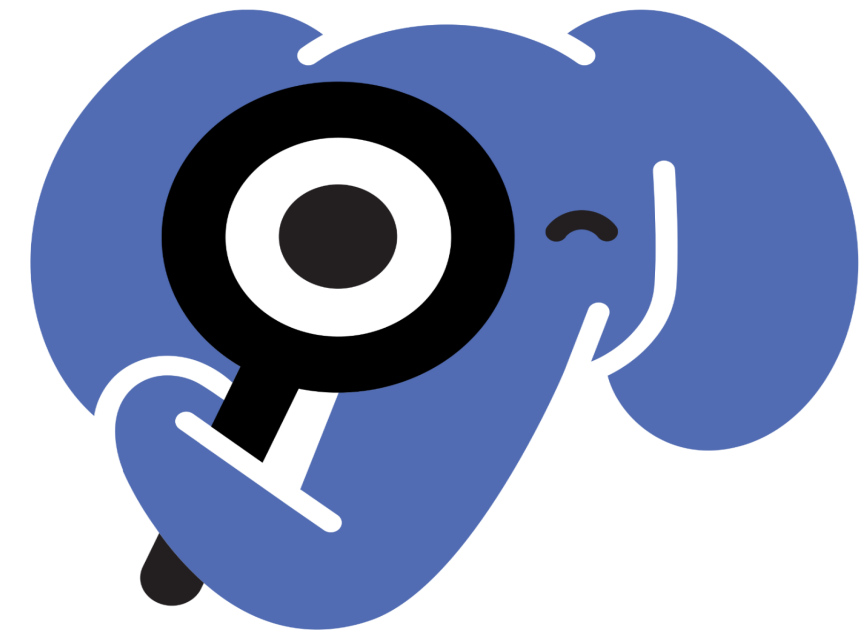
```
parameters:
  drupal:
    entityMapping:
      node:
        class: Drupal\node\Entity\Node
        storage: Drupal\node\NodeStorage

      node_type:
        class: Drupal\node\Entity\NodeType
        storage: Drupal\Core\Config\Entity\ConfigEntityStorage

      shortcut:
        class: Drupal\shortcut\Entity\Shortcut
        storage: Drupal\Core\Entity\Sql\SqlContentEntityStorage

      user:
        class: Drupal\user\Entity\User
        storage: Drupal\user\UserStorage
```

PHPStan



```
<?php

$storage = EntityManager->getStorage('node');
$storage->load($id);
```

Performance

Predictability

Decrease
downtime

What could it bring to compile...

Plugins: Let's focus on FieldType

```
<?php

use Drupal\Core\Field\FieldStorageDefinitionInterface;
use Drupal\Core\TypedData\DataDefinition;

public static function propertyDefinitions(FieldStorageDefinitionInterface $field_definition)
{
    $properties = parent::propertyDefinitions($field_definition);

    $properties['whatever'] = DataDefinition::create('string');

    return $properties;
}
```

26

Testing

PHPStan



What could it bring to compile...

Plugins: Let's focus on FieldType

```
<?php

$link = $this->createMock(LinkItemInterface::class);
$link->method('isEmpty')
    ->willReturn(FALSE);
$link->method('__isset')
    ->with('uri')
    ->willReturn(TRUE);
$link->method('__get')
    ->with('uri')
    ->willReturn($faker->url());

$article = $this->createMock(Article::class);
$article->method('get')
    ->with('field_link')
    ->willReturn($link);
$article->method('isPublished')
    ->willReturn(TRUE);
```

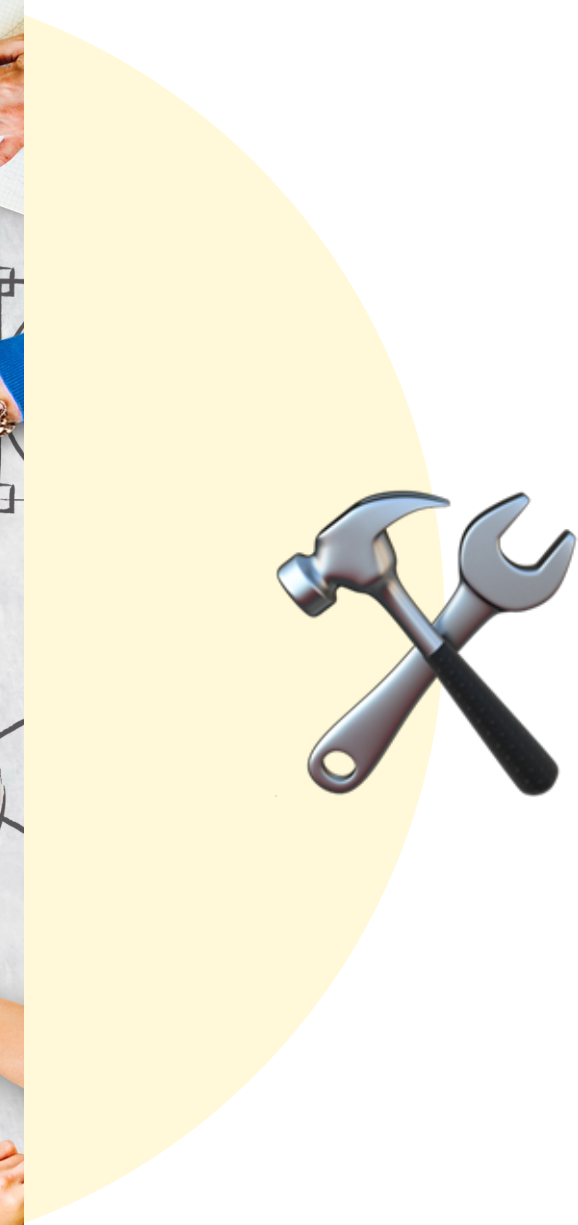


```
<?php

anArticle()
    ->published()
    ->with(aLink());
```



Testing



How? What concretely?

What could we do with Drupal as is?

```
<?php

$settings['php_storage']['cache_container'] = [
  'class' => '\Drupal\Component\PhpStorage\FileStorage',
  'arguments' => [
    [
      'directory' => '/someplace/outside/docroot/php',
      'bin' => 'container',
    ],
  ],
];

$settings['bootstrap_container_definition'] = [
  'parameters' => [],
  'services' => [
    // [...]
    'cache.backend.php_factory' => [
      'class' => '\Drupal\Core\Cache\PhpBackendFactory',
      'arguments' => ['@cache_tags_provider.container'],
    ],
    'cache.container' => [
      'class' => '\Drupal\Core\Cache\PhpBackend',
      'factory' => ['@cache.backend.php_factory', 'get'],
      'arguments' => ['container'],
    ],
  ],
];
```

**What could we do with
Drupal as is?**



**We do not control
when the container
will be rebuild:
it is public**

What could we do with Drupal as is?

```
$content = sprintf(
    '<?php return unserialize(%s);',
    var_export(serialize($container), TRUE)
);
```

```
/** @var object{
 *   'cid': string,
 *   'data': array{
 *     'aliases': array<string, string>,
 *     'parameters': array<string, mixed>,
 *     'services': array<string, string>,
 *     'frozen': bool,
 *     'machine_format': bool,
 *   },
 *   'expire': int,
 *   'tags': array,
 *   'checksum': int
 * } $container
 */
```

Generate some parts from a Drupal instance

Optional config for...

PHPStan



Container

Entity Definition:
Storage...

Routes

Plugin

Etc...

Know the distinctive characteristics for each items

```
/** @var object{
 *   'cid': string,
 *   'data': array{
 *     'aliases': array<string, string>,
 *     'parameters': array<string, mixed>,
 *     'services': array<string, string>,
 *     'frozen': bool,
 *     'machine_format': bool,
 *   },
 *   'expire': int,
 *   'tags': array,
 *   'checksum': int
 * } $container
 */
```



Private services are removed
from the container
during compilation

An alternative Kernel?

Why not?

34



Big picture

Load services from settings

Load core/core.services.yml

Read core.extension.yml

Load extensions' services

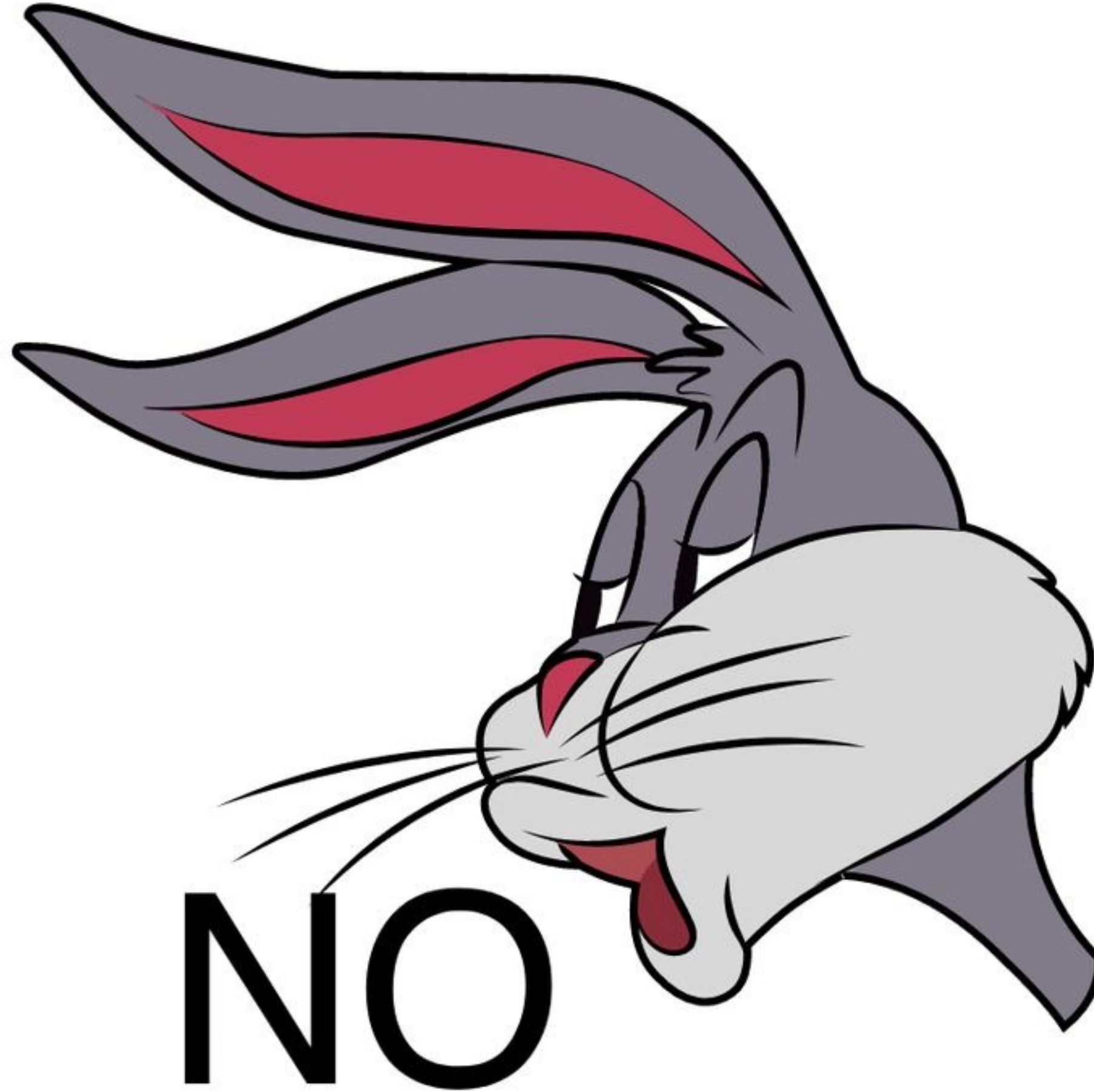
Go handle Compiler Passes

And a lot more...





Is it only a technical solution/approach?





Together



Some links

<https://www.12factor.net/>

<https://www.jrockowitz.com/blog/deployment-with-minimal-downtime>

<https://www.reddit.com/r/drupal/comments/a8b71o/comment/ec9d6jv/>

<https://ewaldbenes.com/blog/guidelines-for-database-changes-with-zero-downtime>

<https://phauer.com/2015/databases-challenge-continuous-delivery/>

<https://www.drupal.org/project/drupal/issues/2497243#comment-10193543>

<https://www.drupal.org/project/drupal/issues/3397224>

<https://www.drupal.org/project/drupal/issues/2301163>

<https://www.drupal.org/project/drupal/issues/2531564>

<https://www.drupal.org/node/3327856>

<https://www.drupal.org/project/drupal/issues/2544932>

<https://trunkbaseddevelopment.com/>

<https://philsturgeon.com/api-evolution-for-rest-http-apis/>



Merci
pour votre
écoute !

Any Questions?

