



Queue, our esteemed utility



Brad Czerniak

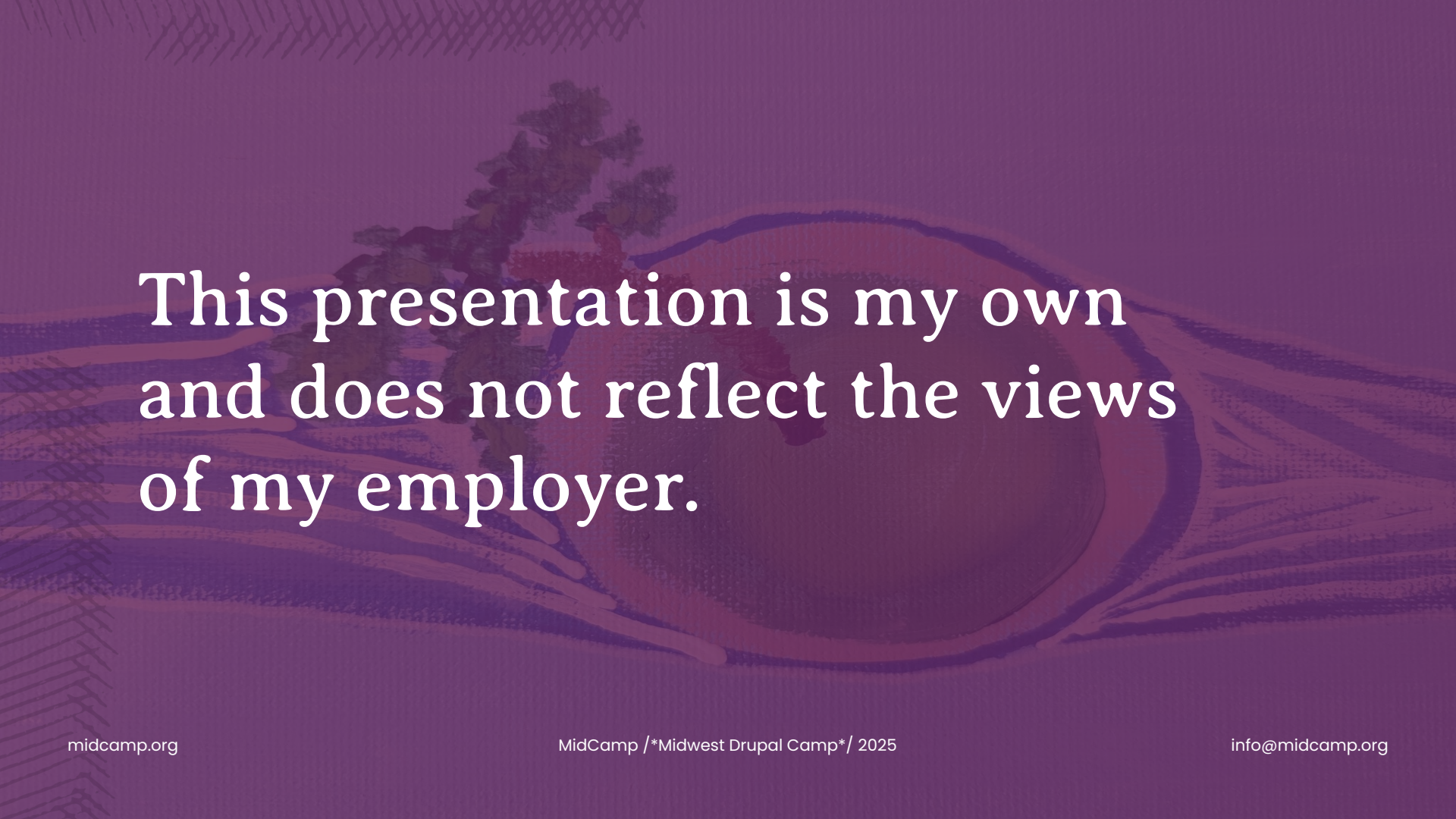
Senior Software Engineer
at Principal Financial Group
@ao5357

First, an apology:

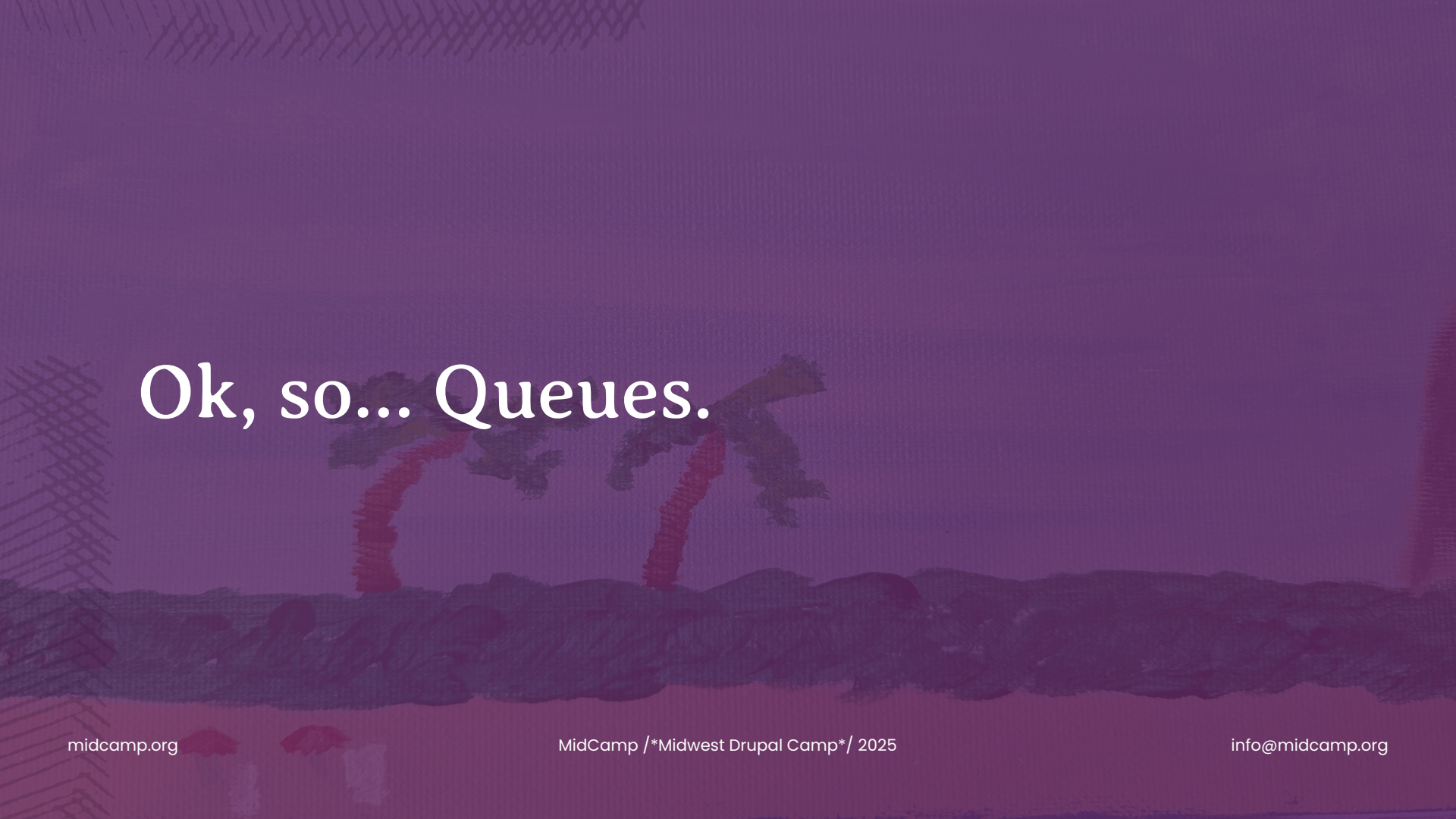
Sorry, I have a big distraction
planned.



While I'm talking, a lovely
volunteer will give away the
slide backgrounds.



This presentation is my own
and does not reflect the views
of my employer.



Ok, so... Queues.

Things you might want to do:

- Make new content
- Update existing content
- Tweak config
- Delete stuff
- Sync with a third party
- Change one thing to another
- Manage revisions, translations, paths, etc.

Ways to do stuff:

- Manually via the UI
- Queues! (and batches)
- Migrations (and Feeds)
- cron, post_update, hooks
- Core and contrib modules
 - default_content
 - views_bulk_operations
 - ECA or rules
- Other custom coding patterns



Quick pause: What's your problem?

How does a *well-to-do*
person-about-town decide?

Let's play “Should we queue
it?”

Should we queue it #1:

Install updated yaml from a
custom module's
config/optional directory

Nope!

This is where `post_update` hooks shine. Check out Dave Reid's [helper module](#) to make it even easier.

Should we queue it #2:

You added a new checkbox field to articles and you need existing articles to take the 'checked' state.



Nope!

If you can use views bulk operations
and views bulk edit you'll be done,
without custom code, inside an hour.

& gnarly

Should we queue it #3:

Your designer and writer have
provided new mocks and copy
for 55 new pages

Nope!



Time to roll up your sleeves (or
go to an AI session?)

Just smoke it



Should we queue it #4:

Make 1000 taxonomy terms
from a CSV



Yes!

You could also do a migration
or feeds, but there's more to
think about.

Should we queue it #5:

Update 1000 nodes from a CSV

Yes!

If your team prefers editing in spreadsheets, you can export a CSV via Views and re-import the updated file in a quick QueueWorker

Should we queue it #6:

Move content from one field to another for every node on the site

Maybe!

A QueueWorker can definitely do it. If you're tricky with `views_bulk_edit` and tokens maybe you can get away without code.

Should we queue it #7:

Turn an outdated layout builder block into a new section instead, and convert the content into multiple blocks

Yes, thrice yes!

Messing with Layout Builder is
a fantastic QueueWorker use
case.



Should we queue it #8:

Move content from a Drupal 7
site to a new Drupal 11 site.

Probably not!

When making a new site and moving a lot of content, Core migrate (and migrate_tools and migrate_plus) is probably the right answer

Should we queue it #9:

Move over 50 pages of content
from another site with the
same content type.





Nope!

Gotta love default content for
wild stuff like that.

Should we queue it #10:

You have 3 fields on each page
and want to combine their
values for display.

No!

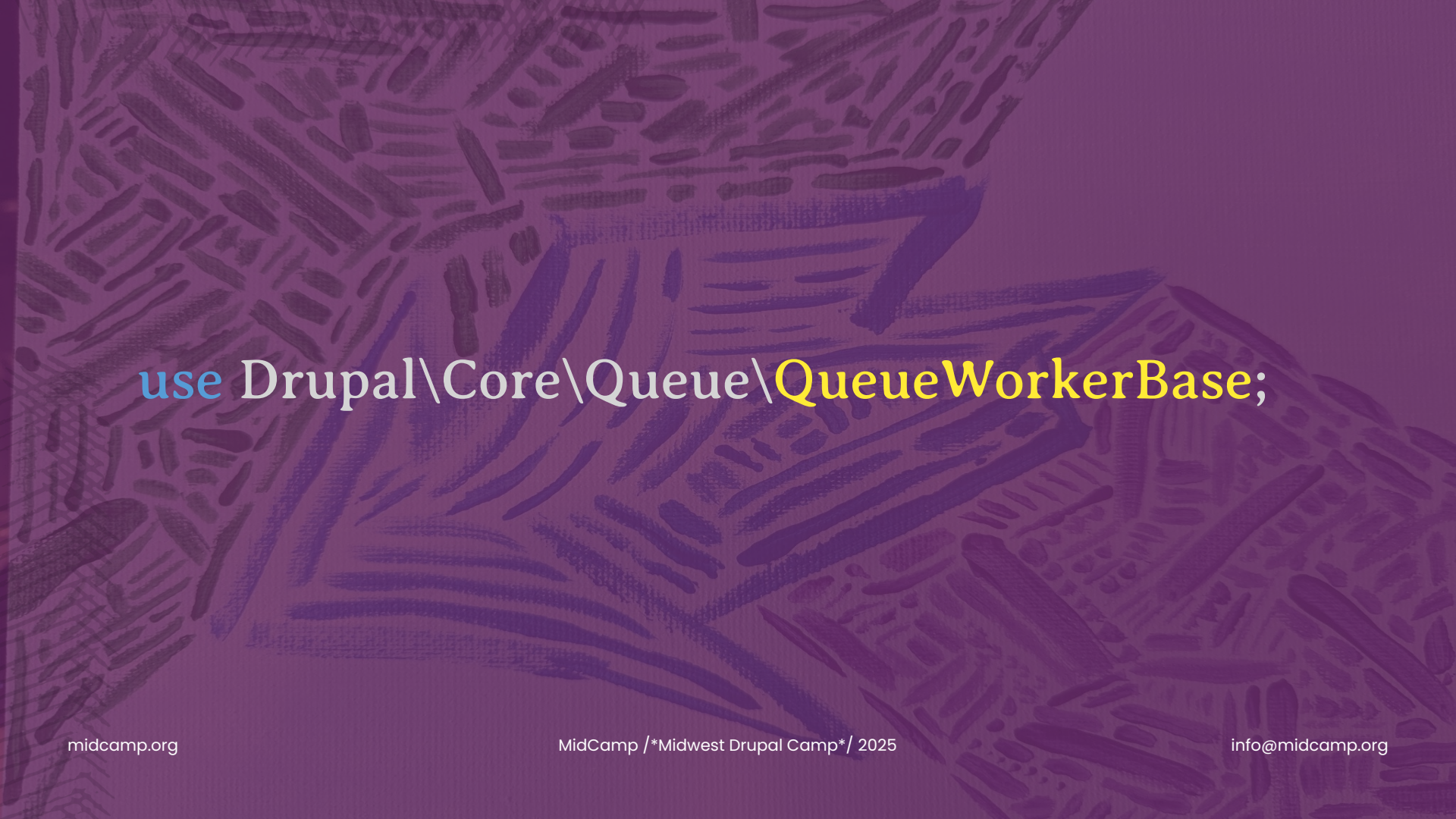
Chances are you want a pseudo-field, a field formatter, field_group + inline CSS, or some twig templating.

Queues shine when:

- You're a programmer
- Working on an existing site
- Total run time over 1 min.
- No need for a GUI
- *There's conditional logic

Queues are less useful for:

- Less-technical folks
- Quick and simple operations
- Full migrations



`use Drupal\Core\Queue\QueueWorkerBase;`



The secret sauce for queues: drush commands

```
/**
 * Queue up each row from a CSV to be processed in a queue worker.
 */
#[Command(name: 'queue-toolbox:create-from-csv', aliases: ['qtcfc'])]
#[Usage(name: 'drush queue-toolbox:create-from-csv', description:
'Queues up the rows')]
public function createFromCsv() {
    $total = 0;

    $queue = $this->queueFactory->get('queue_toolbox_create_from_csv');

    $rows = array_map('str_getcsv', file(realpath(dirname(__FILE__)) .
'../../../../../csv/products-1000.csv'));
}
```



```
// Get the entity bundle key for the query.
$definition = $this->entityTypeManager->getDefinition($entity_type);
$bundle_key = $definition->getKey('bundle');

// Do an entity query with optional bundle.
$storage = $this->entityTypeManager->getStorage($entity_type);
$query = $storage->getQuery()->accessCheck(FALSE);

if (!empty($options['bundle']) && mb_strlen($options['bundle'])) {
    $query->condition($bundle_key, $options['bundle']);
}

$result = $query->execute();
```

36 slides in, an actual piece of a QueueWorker:

```
public function processItem($data) {  
    // Timestamp to pass to prepareRevision().  
    $time = time();  
  
    /** @var \Drupal\Core\Entity\ContentEntityStorageBase $storage  
    */  
    $storage =  
    $this->entityTypeManager->getStorage($data['entity_type']);
```

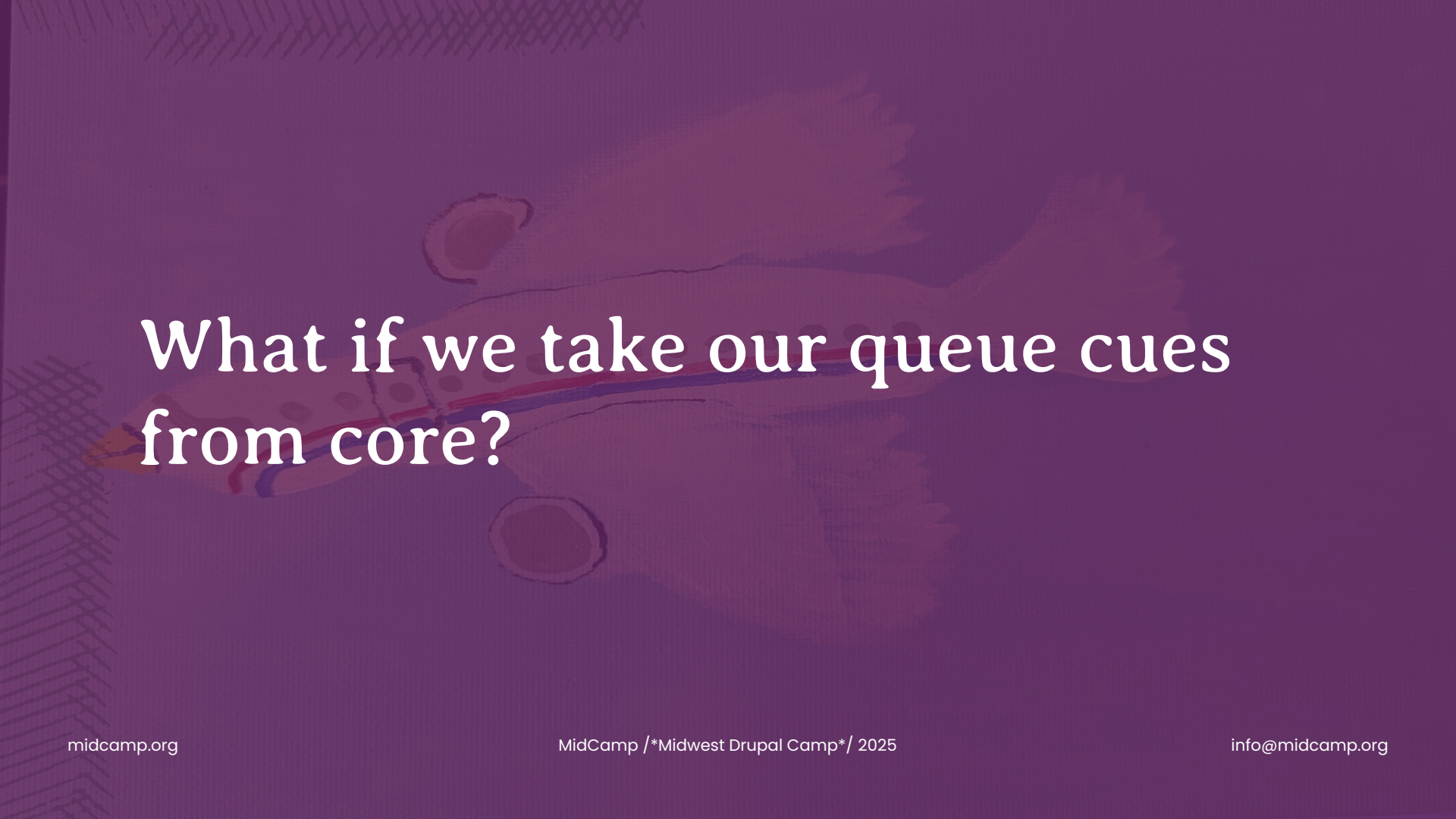



Lesson learned: there's a right way and many wrong ways to deal with revisionable entities

```
// First load the entity.
$entity = $storage->load($data['entity_id']);
if (!empty($entity)) {
    if ($entity instanceof RevisionableInterface && !$entity->isLatestRevision()) {
        $vid = $storage->getLatestRevisionId($entity->id());
    }

    // Prepare and update current revision of entity.
    $this->prepareRevision($entity, 'Convert hero block to hero section in place.', $time);
    $this->updateEntity($entity, $data);

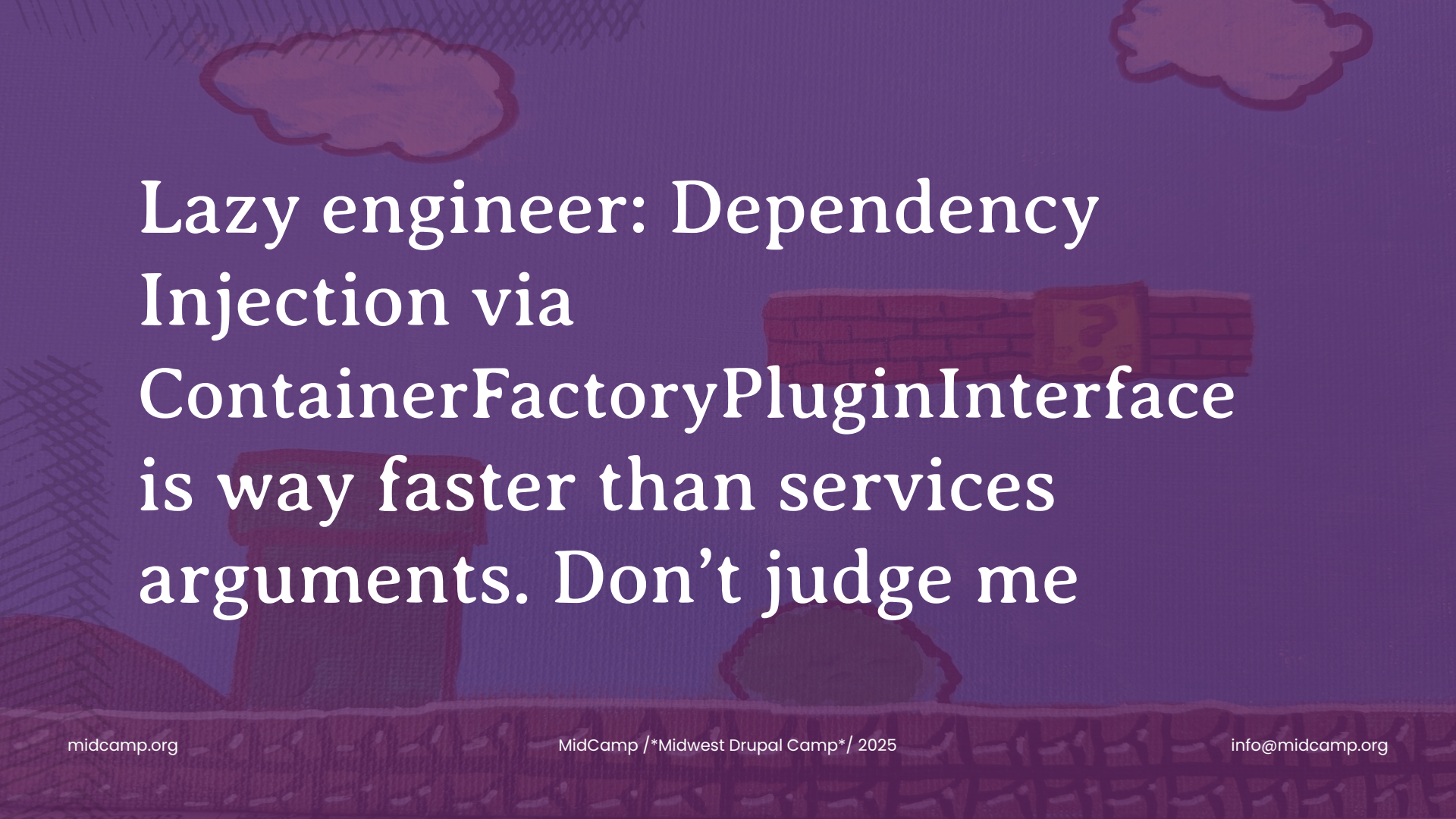
    if (!empty($vid)) {
        /** @var \Drupal\Core\Entity\FieldableEntityInterface $entity */
        $latest = $storage->loadRevision($vid);
        // Do the same to latest, but a millisecond later.
        $this->prepareRevision($latest, 'Convert hero block to hero section in place.', $time +
1);
        $this->updateEntity($latest, $data);
    }
}
}
```

What if we take our queue cues from core?



“But Brad, what about the
batch API?”



Lazy engineer: Dependency
Injection via
`ContainerFactoryPluginInterface`
is way faster than services
arguments. Don't judge me

Migrations: Expectation:



Migrations: Reality:

```
188 description:
189 body/format: constants/full_html
190 field_event_end:
191 -
192   plugin: concat
193   source:
194     - EndDate
195     - EndTime
196   delimiter: ' '
197 -
198   plugin: format_date
199   from_format: 'n/j/Y g:i A'
200   to_format: 'Y-m-d\TH:i:s'
201 field_event_start:
202 -
203   plugin: concat
204   source:
205     - StartDate
206     - StartTime
207   delimiter: ' '
208 -
209   plugin: format_date
210   from_format: 'n/j/Y g:i A'
211   to_format: 'Y-m-d\TH:i:s'
212 field_event_type:
213   plugin: entity_generate
214   source: CalendarName
215   entity_type: taxonomy_term
216   bundle_key: vid
217   bundle: event type
```

Bonus stuff in sample module:

- Term ID by name helper
- Example of pathMatcher
- Get node by path alias helper
- Layout Builder toArray tricks



Get the code!

https://github.com/solve-it-once/queue_toolbox

Quick contrib note: Have a
need for a queue UI or
advanced queue features?
Check out those modules

PLEASE PROVIDE YOUR FEEDBACK!

mid.camp/8713

**THANK YOU,
QUESTIONS?**



CONTRIBUTION DAY

Thursday 10:00am to
4:00pm

You don't have to know code to give back!

New Contributor training 9:00am to 10:00am