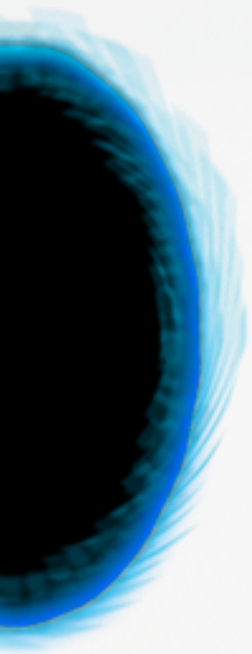
# Using Queues and Offline Processing

David Stockton

Madison PHP - Oct 1, 2016

# Synchronous Processing



# Shopping Trip

# Car Wash

DMV

# The Web Should Be Fast

# Customers Want It To Be Fast

# Think Work Order

# Asynchronous Processing

# Real-World Async

# Online Shopping

# Personal Assistants

# Personal Assistants

# Web Pages

# Infinite Scrollers

# Magical Parallelization

# Fake It Until You Make It

# Time Is Not On Our Side

# User Input

# Do The Needful Now
# Everything Else Can Wait

# Database Queue

# Data Synchronization

# Message Queues

# Work Producer

Work

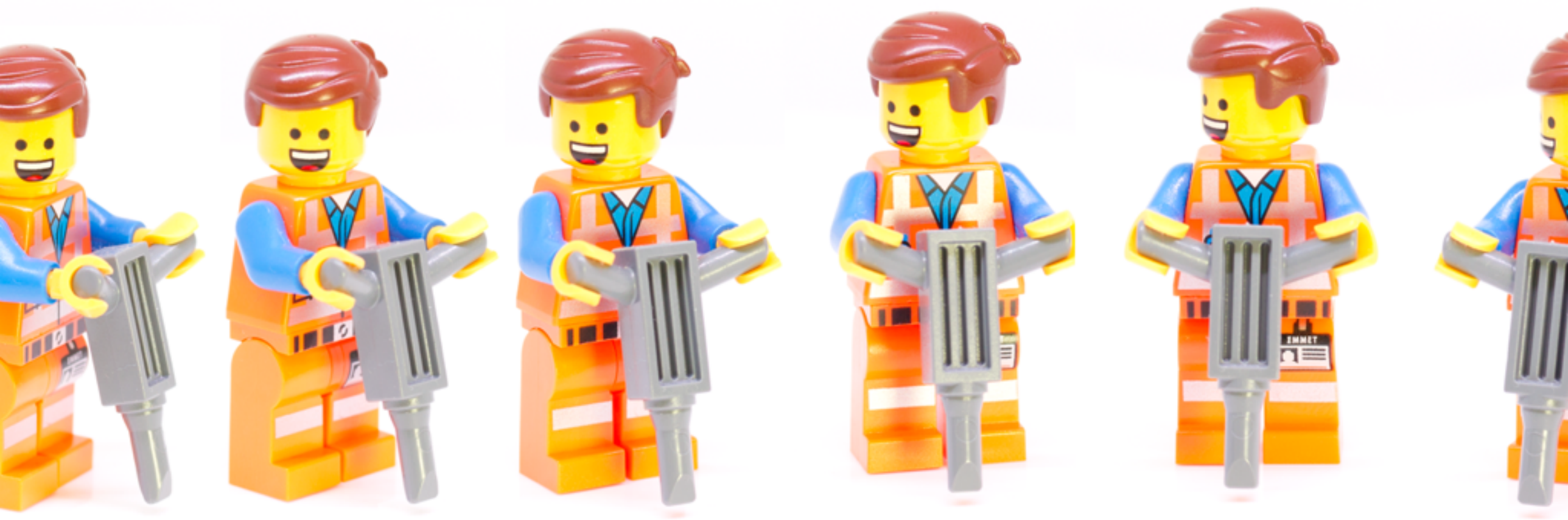
Producer

# How does that help?

# Workers

- Workers

# Multiplicity



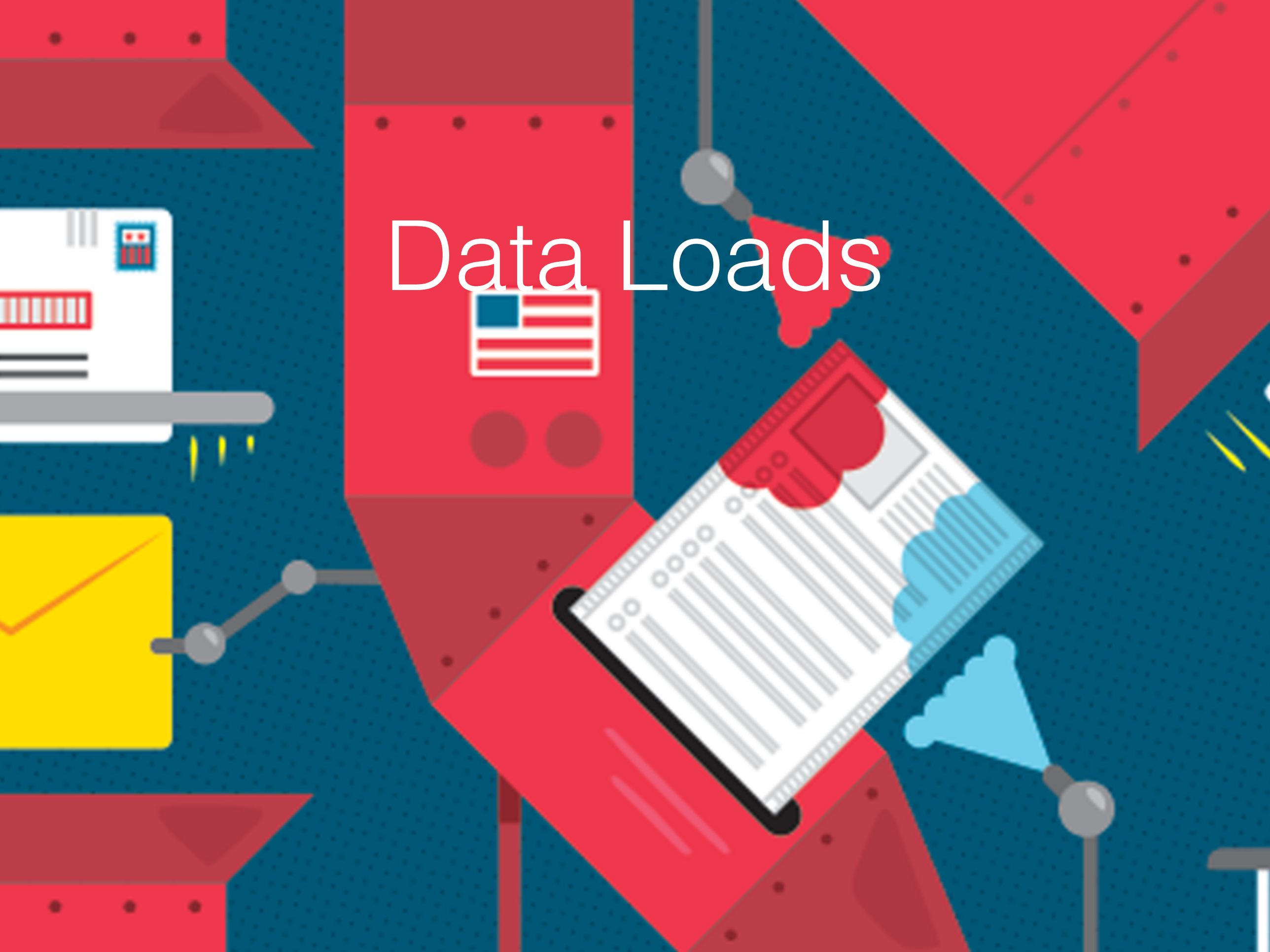
# Control Server Load

# Handle Load Spikes

# Workers as Assistants

# Ballottrax

# Data Loads



# Communication Rates

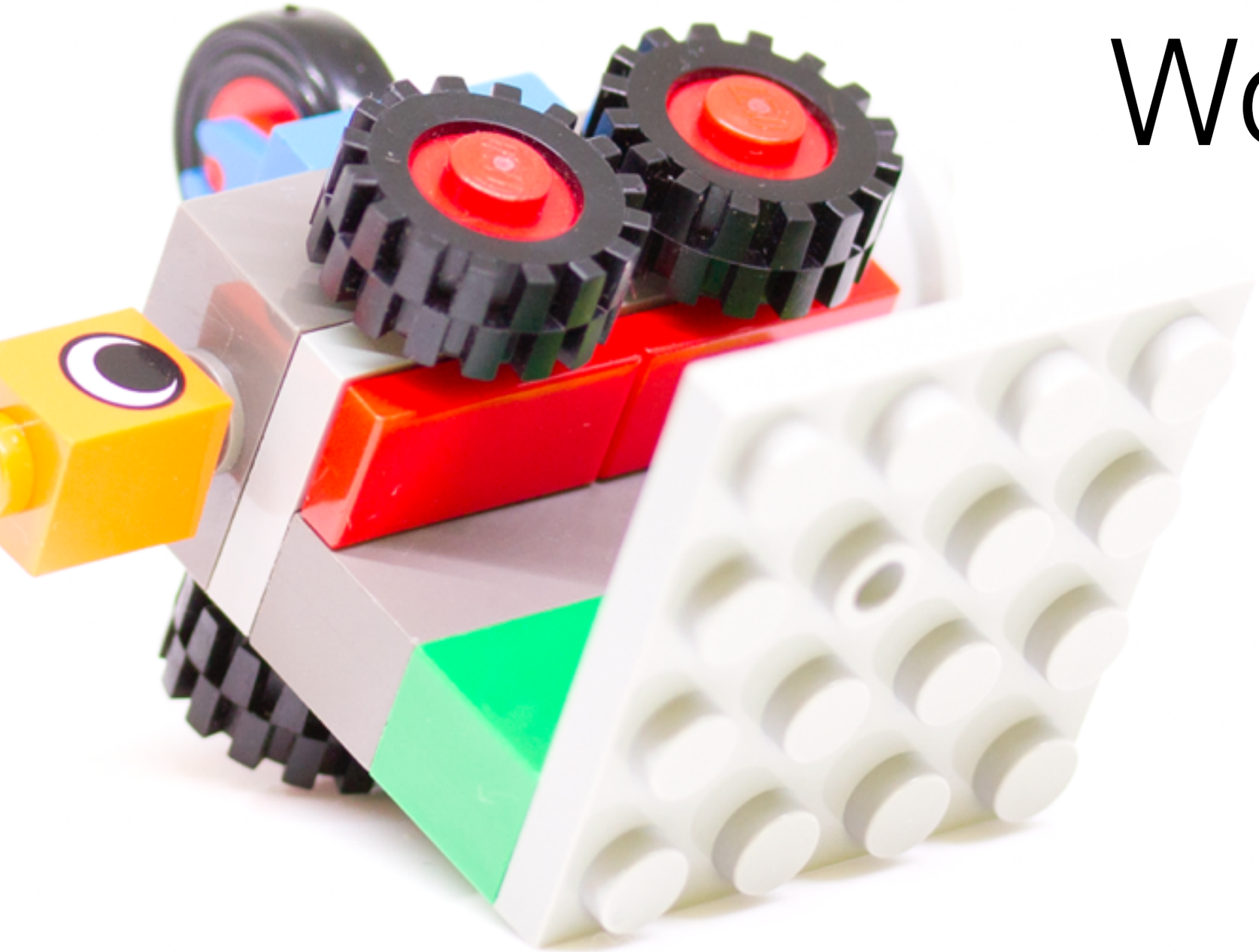
- Email - 250 connections, 10 per seco
- SMS - 30 per second
- Voice - 1 per second, per phone number

# Splitting Work By Type

# Rate Limiting

# Queues to Queue Work

# Acceptable for Work



# Notification Filters

# Geocoding

Producers → Consumers

# DB Queue vs Pure MQ

# Feeding Queue from Database - Cron

# Feeding Queues from Database - Polling

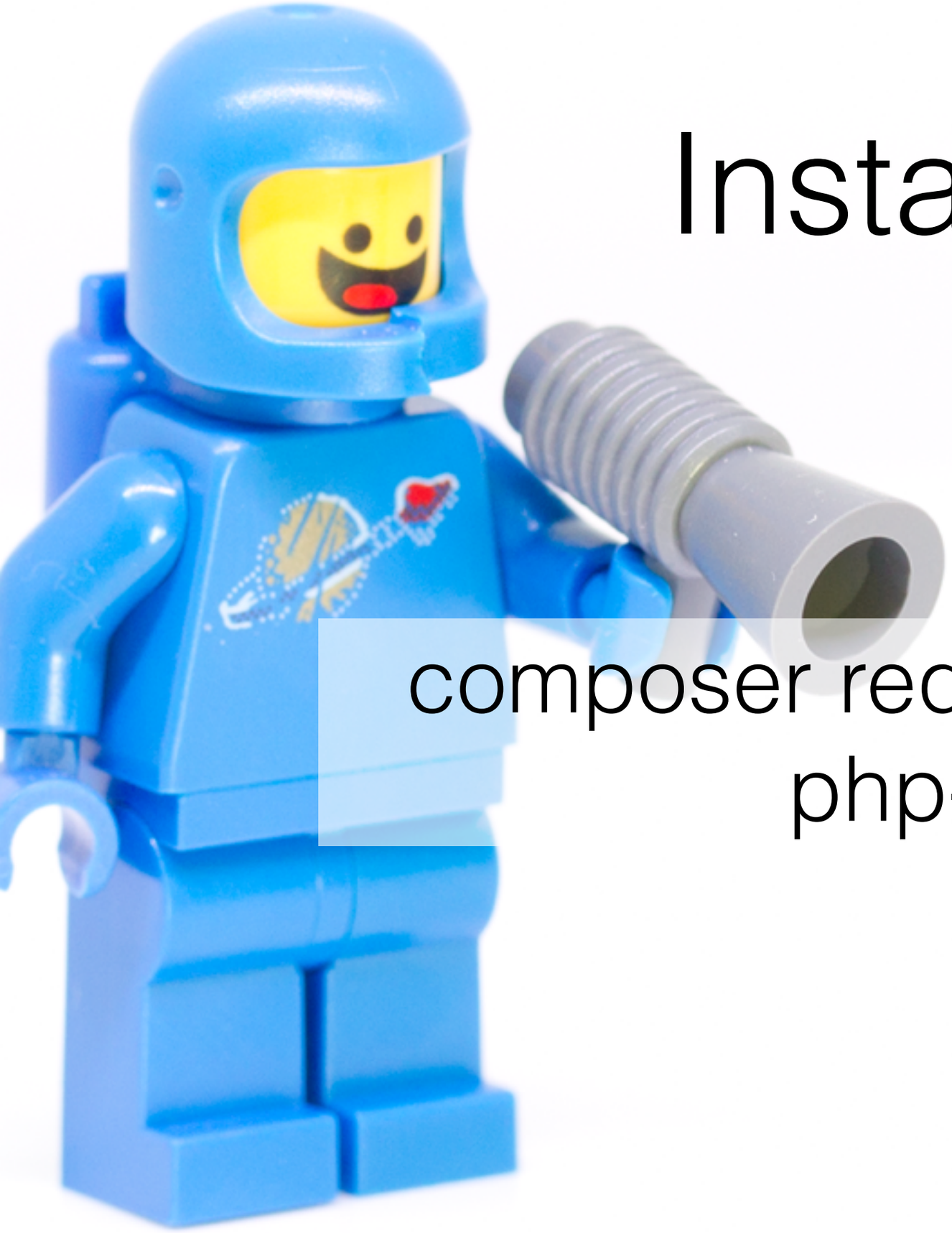
# Polling - The Worst

# Events and Webhooks

# Magical Webhooks

# Automated Pull Requests

# Queues For Webhooks



# Install AMQPLib

composer require php-amqplib/
php-amqplib

# Create Queue/Channel

```
$connection = new \PhpAmqpLib\Connection
\AMQPStreamConnection($server, $port, $user,
$password, $vhost);
$channel    = $connection->channel();
$channel->queue_declare(
    $queueName,
    false, // Passive
    true,  // Durable
    false, // Exclusive
    false  // Auto Delete
);
```

# Setting Channel Options

```
$channel->basic_qos(null, 1, null);
$channel->basic_consume(
    $queueName,
    '', // Consumer tag
    false, // No local
    false, // No ACK
    false, // Exclusive
    false, // No wait
    $callback
);
```

# Do Some Work

```
while (count($channel->callbacks)) {
    $channel->wait();
}

$channel->close();
$connection->close();
```

# The Callback

```
$callback = function ($msg) {
    $message = (array) json_decode($msg-
>body, false);
    // Do work

    $msg->delivery_info['channel']
->basic_ack(
        $msg->delivery_info['delivery_tag']
    );
};
```

# Creating a Message

```
$message = new \PhpAmqpLib\Message\AMQPMessage(
    json_encode($data),
    [
        'delivery_mode' =>
            AMQPMessage::DELIVERY_MODE_PERSISTENT
    ]
);
```

# Putting Message in the Queue

```
try {
    $channel->basic_publish($message, '',
$queueName);
} catch (AMQPExceptionInterface $e) {
    $this->getLogger()->err(
        'Error putting message into RabbitMQ',
        $e->getMessage()
    );
    return false;
}
```

# Work Priority

# Recap

# Questions?

David Stockton
@dstockto
https://davidstockton.com
https://tddftw.com