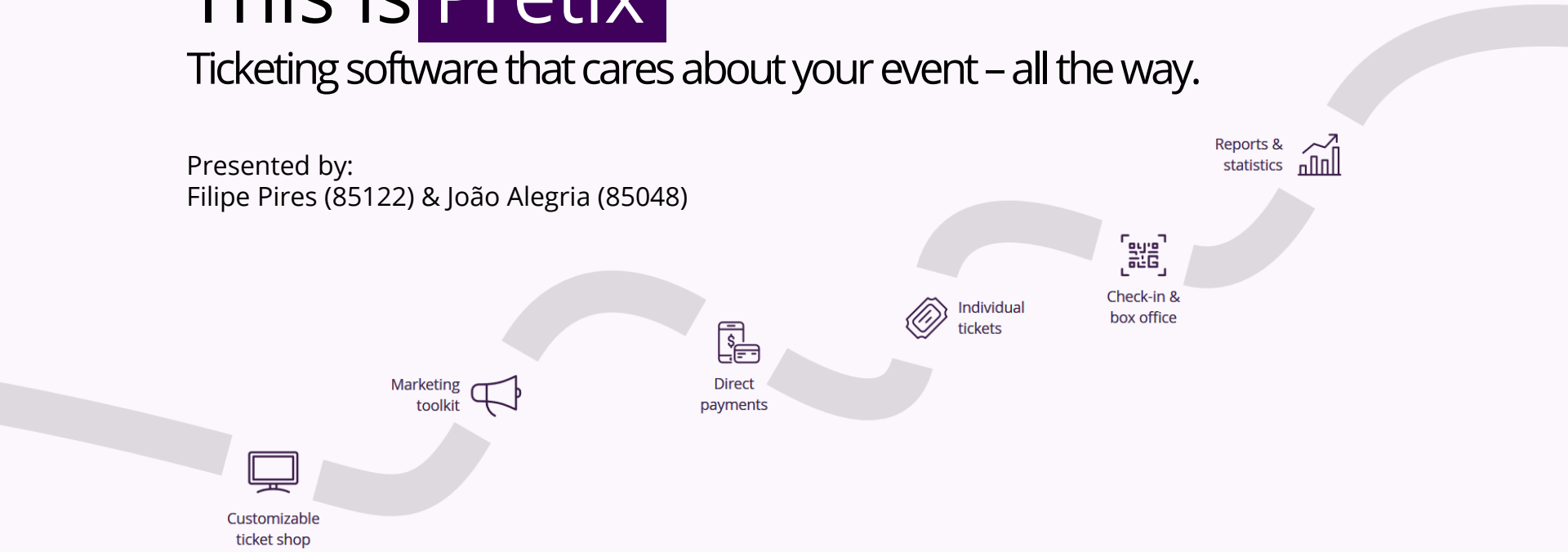
This is Pretix

Ticketing software that cares about your event – all the way.

Presented by:
Filipe Pires (85122) & João Alegria (85048)



1

Supplied Product

What will we deploy and manage exactly ?

Pretix as our Product

- All-in-one: Online shop, box office and ticket outlets
- Focus on privacy and security
- Extensible with plug-ins and through a REST API

Successfully used for conferences, festivals, concerts, shows, exhibitions, workshops, barcamps, and more.

2

Technical Aspects

What software ? What is our purpose ? How will it be used ?

Deployment & Support

Dependencies

PostgreSQL database server for persistence.

Redis server for cache.

NGinX proxy for HTTPS connections.

Docker compose for deployment.

Our Purpose

To deploy Pretix and host a tech event (like WebSummit).

To ensure the system is capable of responding to a high number of ticket purchases.

Usage

Users will be generated and controlled programmatically through Pretix's REST API.

Logging and other maintenance techniques will be used to monitor the entire infrastructure.

3

Distributed Execution

What will our method be ?

Distribution & Availability

- Database replication
- Deployment of 2 instances of each module
- Usage of an external proxy for load distribution

Considering the limitations provided by the creators of Proxy, we hope to successfully respond to 800 ticket purchase requests to the same event per minute.

Our focus is on providing Pretix for the portuguese tech community.

We intend to demonstrate this by hosting a simulated large technology event.

To do this, we aim to deploy 2 instances of each module with a memory requirement of 8Gb.

4

Example Use Case

Just to give you a better notion ...

Simple Usage Example

- Generate thousands of ficticious authenticated users
- Simulate different loads of simultaneous ticket purchases
- Test how the system responds to local failures
- Analyse results and report conclusions to deal with potential issues

Our tech event use case is meant to illustrate a realistic and appropriate usage scenario where a deployed system such as Pretix must be fault tolerant and scaled to the dimension of the event itself.

5

Work Plan & Execution

Where will we start from ?

Project Planning

1. Deploy all components in individual docker containers
2. Ensure correct functioning of the whole system and basic features
3. Distribute service availability while ensuring data integrity
4. Develop strategical measures for traffic control

> Test and evaluate solutions during the entire workflow

Thanks!

Any questions ?