

# iPhone Push bei den VZ Netzwerken

Max Horváth, Steffen Irrgang





# Vorstellung der Speaker





# Was sind Push Notifications?



# Was sind Push Notifications?

- Eine Möglichkeit, eine Nachricht an individuelle Nutzer einer iPhone App zu senden



# Was sind Push Notifications?

- Eine Möglichkeit, eine Nachricht an individuelle Nutzer einer iPhone App zu senden
- Vergleichbar mit SMS Nachrichten, sie werden allerdings nicht im Telefon abgespeichert





# Was sind Push Notifications?

- Eine Möglichkeit, eine Nachricht an individuelle Nutzer einer iPhone App zu senden
- Vergleichbar mit SMS Nachrichten, sie werden allerdings nicht im Telefon abgespeichert
- Transportiert über einen sicheren Kommunikationsweg





# Was sind Push Notifications?

- Leichtgewichtig, um sehr hoch zu skalieren





# Was sind Push Notifications?

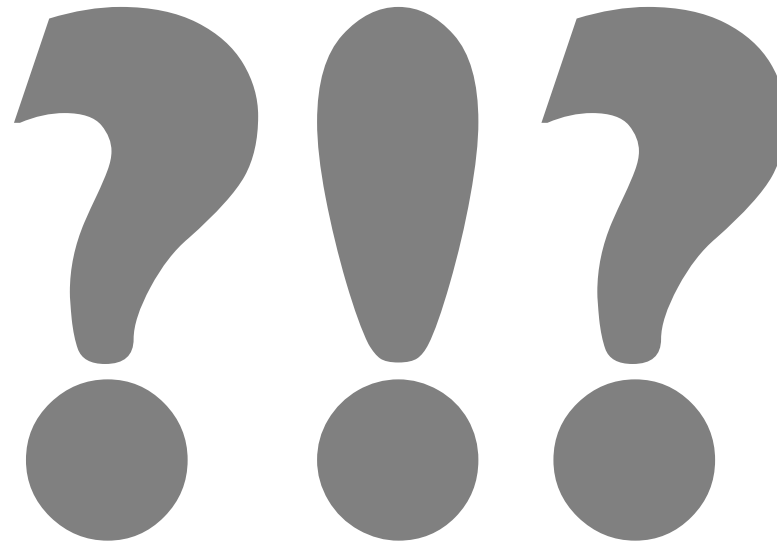
- Leichtgewichtig, um sehr hoch zu skalieren
- Kostenlos zu senden und zu empfangen







# Was sind sie nicht?



# Was sind sie nicht?

- Ein Hintergrunddienst für eine iPhone App

# Was sind sie nicht?

- Ein Hintergrunddienst für eine iPhone App
- Ein vollständiger Ersatz für einen Hintergrunddienst einer iPhone App

# Was sind sie nicht?

- Ein Hintergrunddienst für eine iPhone App
- Ein vollständiger Ersatz für einen Hintergrunddienst einer iPhone App
- Ein Kanal, um Daten zu einer iPhone App zu transportieren (Limitierung auf 256 Bytes)



# Wie werden sie übertragen?





# Wie werden sie übertragen?





# Wie werden sie übertragen?





# Wie werden sie übertragen?





# Wie werden sie übertragen?



1. App registriert sich bei Apple



# Wie werden sie übertragen?



2. App  
überträgt  
Device ID



# Wie werden sie übertragen?



3. Übertragung einer Nachricht an Apple



# Wie werden sie übertragen?



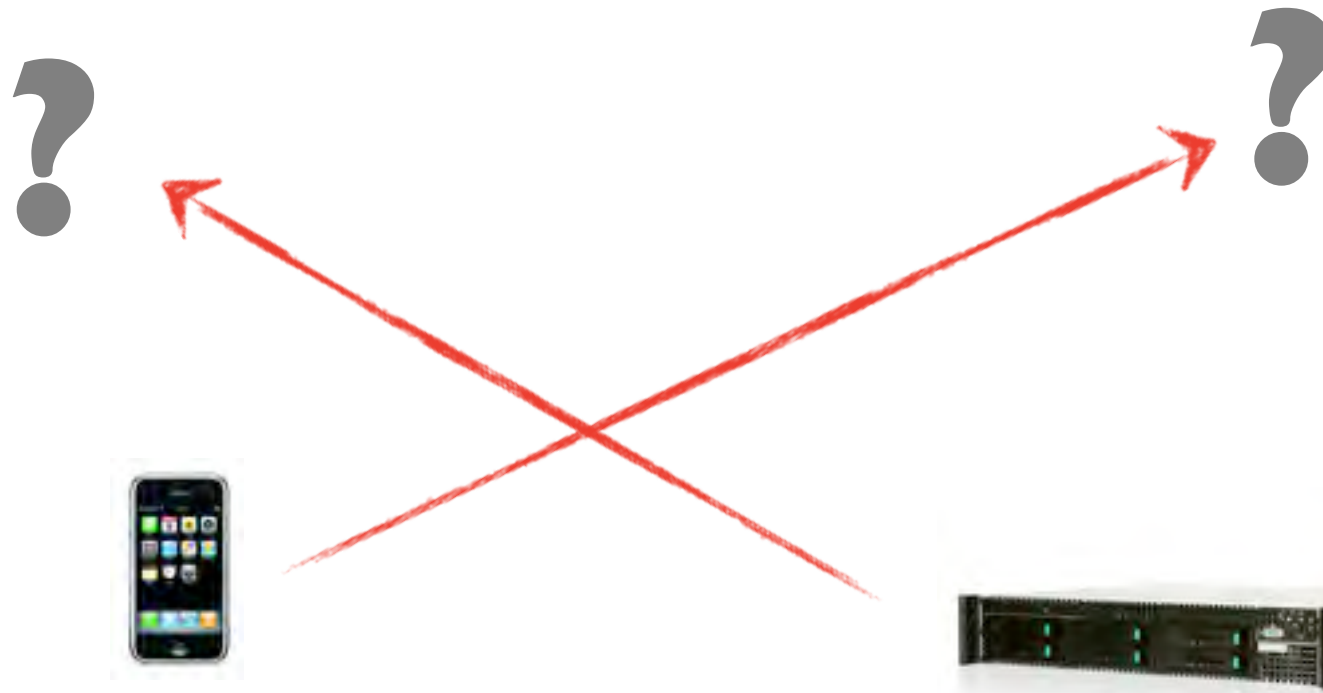
4. Apple sendet  
Nachricht an iPhone



# Wie werden sie übertragen?



# Auf Fehler reagieren



# Auf Fehler reagieren

- Bei der Übertragung von Push Notifications kann es zu verschiedenen Fehlern kommen.

# Auf Fehler reagieren

- Bei der Übertragung von Push Notifications kann es zu verschiedenen Fehlern kommen:
  - Das iPhone des Users ist für längere Zeit offline



# Auf Fehler reagieren

- Bei der Übertragung von Push Notifications kann es zu verschiedenen Fehlern kommen:
  - Das iPhone des Users ist für längere Zeit offline
  - Der User hat die App wieder deinstalliert

# Auf Fehler reagieren

- Bei der Übertragung von Push Notifications kann es zu verschiedenen Fehlern kommen:
  - Das iPhone des Users ist für längere Zeit offline
  - Der User hat die App wieder deinstalliert
  - Man sendet Apple eine ungültige Device ID



# Auf Fehler reagieren

- Auf diese Fehler muss man reagieren, da Apple einen sonst von der Nutzung des APNS ausschließen kann



# Auf Fehler reagieren

- Auf diese Fehler muss man reagieren, da Apple einen sonst von der Nutzung des APNS ausschließen kann
- Apple bietet dazu den Feedback Service



# Auf Fehler reagieren

- Auf diese Fehler muss man reagieren, da Apple einen sonst von der Nutzung des APNS ausschließen kann
- Apple bietet dazu den Feedback Service
- Er liefert Device IDs, die Apple nach mehreren Zustellversuchen nicht übertragen konnte.





# Auf Fehler reagieren

- Man sollte den Feedback Service regelmäßig abrufen und übergebene Device IDs aus der eigenen Datenbank entfernen, um keine Nachrichten an diese in Zukunft zu senden.



# Voraussetzungen



# Voraussetzungen

- Möglichst zwei Geräte



# Voraussetzungen

- Möglichst zwei Geräte:
  - Für die Entwicklung (Sandbox-Zugriff)

# Voraussetzungen

- Möglichst zwei Geräte:
  - Für die Entwicklung (Sandbox-Zugriff)
  - Für den Live Betrieb

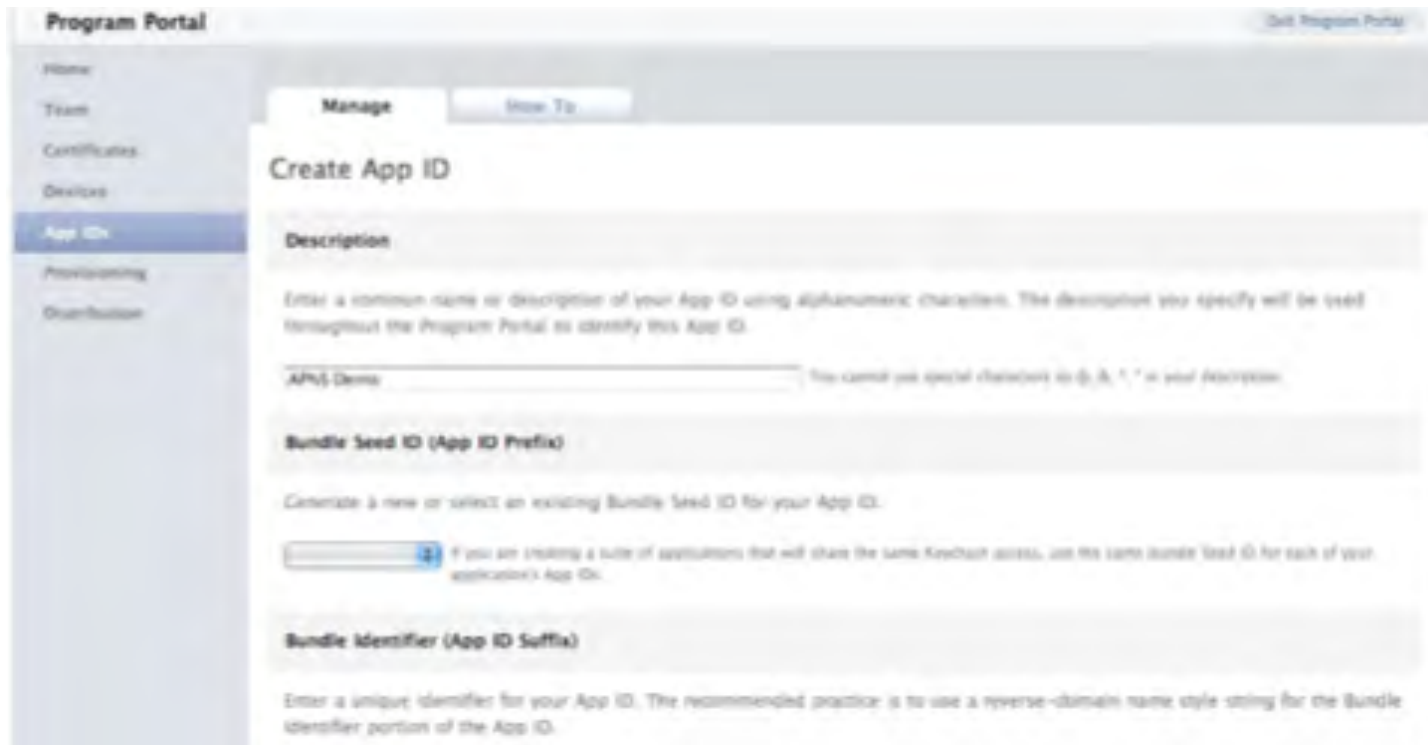
# Voraussetzungen

- Möglichst zwei Geräte
  - Für die Entwicklung (Sandbox-Zugriff)
  - Für den Live Betrieb
- Eigene Serverseite, um Nachrichten an den APNS zu senden

# Voraussetzungen

- Zusätzlich zur iPhone App ID und den Provisions Profilen zwei Zertifikate für die Kommunikation mit dem APNS

# App ID



**Program Portal** Get Program Portal

Home  
Team  
Certificates  
Devices  
**App ID**  
Provisioning  
Distribution

**Manage** Show To

### Create App ID

**Description**

Enter a common name or description of your App ID using alphanumeric characters. The description you specify will be used throughout the Program Portal to identify this App ID.

You cannot use special characters (e.g., &, !, \* ) in your description.

**Bundle Seed ID (App ID Prefix)**

Generate a new or select an existing Bundle Seed ID for your App ID.

If you are creating a suite of applications that will share the same keychain access, use the same bundle seed ID for each of your application's App IDs.

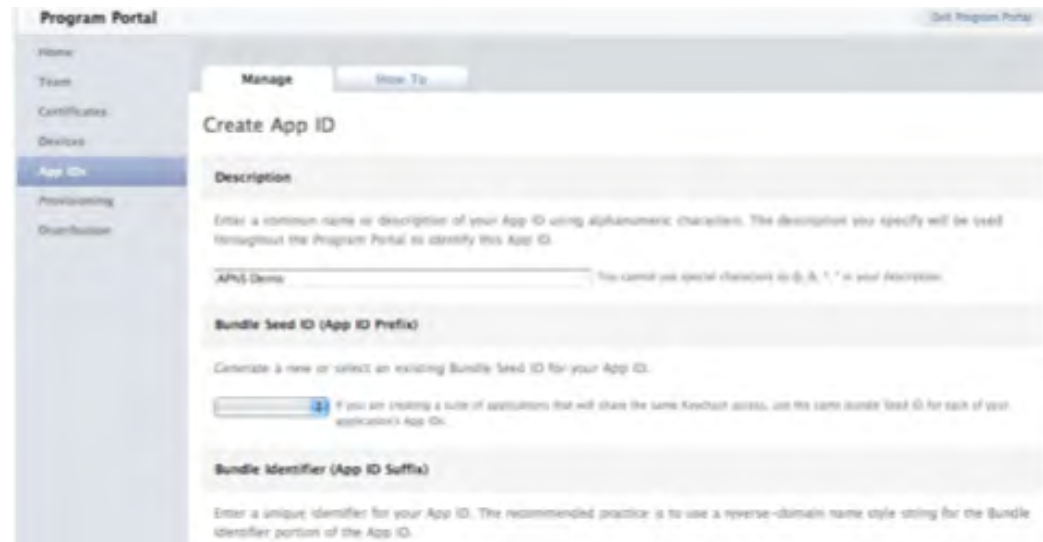
**Bundle Identifier (App ID Suffix)**

Enter a unique identifier for your App ID. The recommended practice is to use a reverse-domain name style string for the Bundle Identifier portion of the App ID.



# App ID

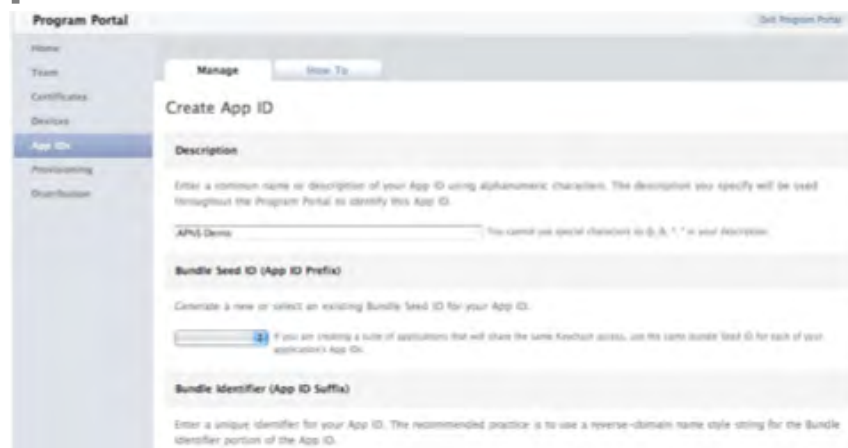
- Wildcards sind nicht erlaubt



The screenshot shows the 'Create App ID' form in the Program Portal. The form is titled 'Create App ID' and has a 'Description' section with a text input field. Below this is the 'Bundle Seed ID (App ID Prefix)' section, which includes a 'Generate' button and a text input field. The final section is 'Bundle Identifier (App ID Suffix)', which has a text input field. The form is part of a larger interface with a sidebar on the left containing navigation links like 'Home', 'Team', 'Certificates', 'Devices', 'App ID', 'Processing', and 'Distribution'.

# App ID

- Wildcards sind nicht erlaubt
- Für bestehende Apps müssen die Provisionsprofile neu erstellt werden



The screenshot shows the 'Create App ID' form in the Program Portal. The form includes a 'Description' field with instructions to use alphanumeric characters. Below it is an 'APNS Domain' field with a note that special characters are not allowed. The 'Bundle Seed ID (App ID Prefix)' section offers a 'Generate a new' button and a 'Select an existing' dropdown. A note explains that the same seed ID should be used for multiple applications. The 'Bundle Identifier (App ID Suffix)' field has instructions to use a reverse-domain name style string.



# Bestehende App IDs

\*.ID



# Bestehende App IDs

- App IDs, die mit einer Wildcard erstellt wurden, müssen nun eine eindeutige ID erhalten

# Bestehende App IDs

- App IDs, die mit einer Wildcard erstellt wurden, müssen nun eine eindeutige ID erhalten:
  - LZI476LW65.\*

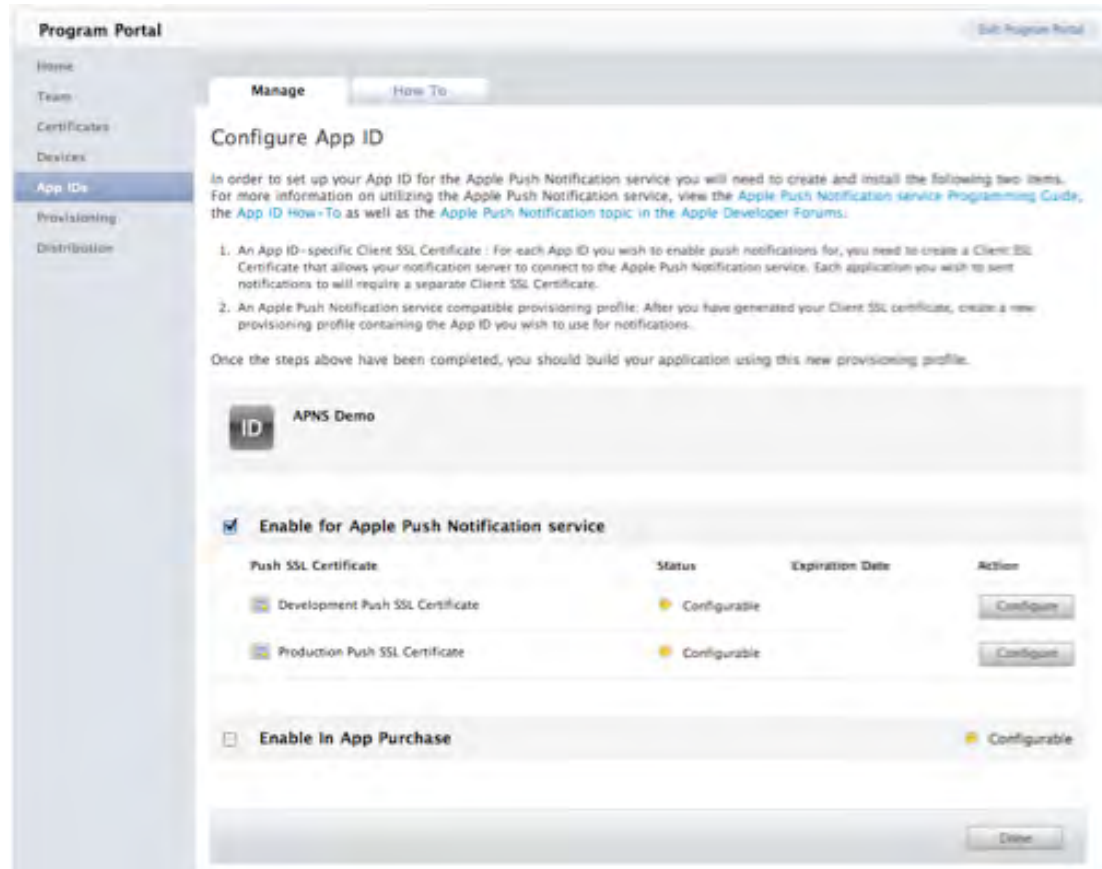
# Bestehende App IDs

- App IDs, die mit einer Wildcard erstellt wurden, müssen nun eine eindeutige ID erhalten
  - LZI476LW65.\*
- Neue App ID inklusive der alten ID erstellen

# Bestehende App IDs

- App IDs, die mit einer Wildcard erstellt wurden, müssen nun eine eindeutige ID erhalten
  - LZI476LW65.\*
- Neue App ID inklusive der alten ID erstellen:
  - LZI476LW65.com.company.app

# Push-Zertifikate



**Program Portal** Sub-Program Portal

Home  
Team  
Certificates  
Devices  
**App IDs**  
Provisioning  
Distribution

**Manage** How To

### Configure App ID

In order to set up your App ID for the Apple Push Notification service you will need to create and install the following two items. For more information on utilizing the Apple Push Notification service, view the [Apple Push Notification service Programming Guide](#), the [App ID How-To](#) as well as the [Apple Push Notification topic in the Apple Developer Forums](#).

1. An App ID-specific Client SSL Certificate : For each App ID you wish to enable push notifications for, you need to create a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification service. Each application you wish to send notifications to will require a separate Client SSL Certificate.
2. An Apple Push Notification service compatible provisioning profile: After you have generated your Client SSL certificate, create a new provisioning profile containing the App ID you wish to use for notifications.

Once the steps above have been completed, you should build your application using this new provisioning profile.

**ID** APNS Demo

**Enable for Apple Push Notification service**

Push SSL Certificate	Status	Expiration Date	Action
Development Push SSL Certificate	Configurable		<input type="button" value="Configure"/>
Production Push SSL Certificate	Configurable		<input type="button" value="Configure"/>

**Enable In App Purchase** Configurable



# Push-Zertifikate

- Müssen für die sichere Kommunikation über SSL erstellt werden

# Push-Zertifikate

- Müssen für die sichere Kommunikation über SSL erstellt werden:
  - Entwicklungszertifikat für die Kommunikation mit der Sandbox

# Push-Zertifikate

- Müssen für die sichere Kommunikation über SSL erstellt werden:
  - Entwicklungszertifikat für die Kommunikation mit der Sandbox
  - Livezertifikat für die Kommunikation mit dem Live-Gateway





# Einbindung in eine iPhone App





# Einbindung in eine iPhone App

- Über `registerForRemoteNotificationTypes:`  
`UIRemoteNotificationType` registriert man sich  
beim APNS





# Einbindung in eine iPhone App

- `didRegisterForRemoteNotificationsWithDeviceToken:deviceToken` liefert die Device ID des iPhones zurück





# Einbindung in eine iPhone App

- `didRegisterForRemoteNotificationsWithDeviceToken:deviceToken` liefert die Device ID des iPhones zurück
- Diese sollte man dann an seinen Server übertragen, damit dieser Nachrichten an den APNS senden kann





# Einbindung in eine iPhone App

- `didFailToRegisterForRemoteNotificationsWithError:error` gibt etwaige Fehler zurück



# Einbindung in eine iPhone App

- Mit `didReceiveRemoteNotification:userInfo` kann man auf Push Notifications reagieren, wenn eine Push Nachricht ankommt, während das Programm läuft

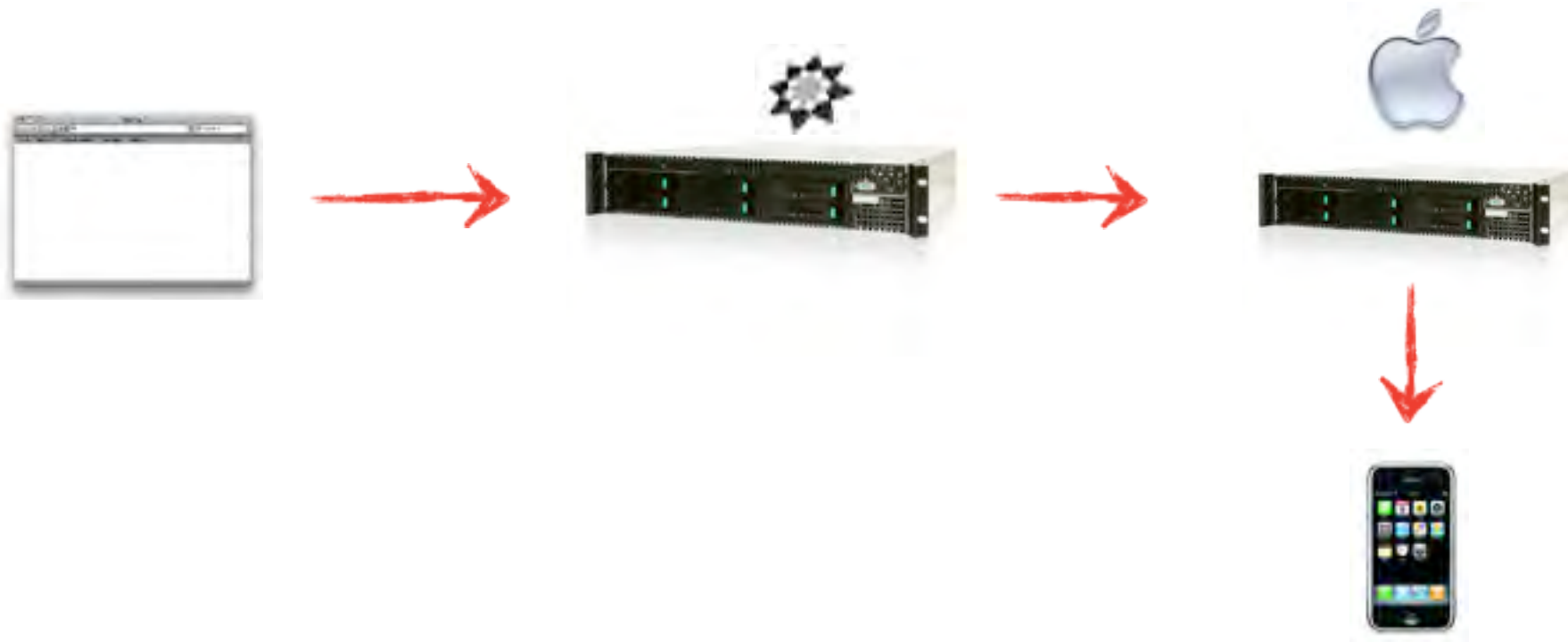




# Client-Server-Kommunikation



# Client-Server-Kommunikation







# Probleme solcher synchronen Third Party Requests





# Probleme solcher synchronen Third Party Requests

- Beispiel bei den VZ Netzwerken





# Probleme solcher synchronen Third Party Requests

- Beispiel bei den VZ Netzwerken:
  - Kommerzielles Profil mit 30.000 Anhängern schreibt eine Nachricht





# Probleme solcher synchronen Third Party Requests

- Beispiel bei den VZ Netzwerken:
  - Kommerzielles Profil mit 30.000 Anhängern schreibt eine Nachricht:
    - Fans wollen per Push, Email und/oder SMS benachrichtigt werden





# Probleme solcher synchronen Third Party Requests

- Beispiel bei den VZ Netzwerken:
  - Kommerzielles Profil mit 30.000 Anhängern schreibt eine Nachricht:
    - Fans wollen per Push, Email und/oder SMS benachrichtigt werden
- Ca. 50.000 Tasks in einem Request!





# Probleme solcher synchronen Third Party Requests

- Ca. 50.000 Tasks in einem Request!





# Probleme solcher synchronen Third Party Requests

- Ca. 50.000 Tasks in einem Request:
  - Timeout?
  - Fehlender Fallback
  - Was passiert, wenn andere Server offline sind?





# Probleme solcher synchronen Third Party Requests





# Offline Processing



# Offline Processing

- Asynchrone Auslagerung
- im „Hintergrund“ arbeiten lassen

# Offline Processing

- Asynchrone Auslagerung
- im „Hintergrund“ arbeiten lassen
- Kontrolle über Responsezeiten

# Offline Processing

- Einfache Möglichkeit: Cronjobs

# Offline Processing

- Einfache Möglichkeit: Cronjobs
  - ausreichend für zyklischen Mailversand
- einfache Zeitsteuerung

# Offline Processing

- Einfache Möglichkeit: Cronjobs
  - ausreichend für zyklischen Mailversand
- einfache Zeitsteuerung
- Beispiel: Zend\_Queue



# Einsatz von Message Queues





# Einsatz von Message Queues

- Bekanntes Problem (z.B. Bankschalter)
- Traditionelles FIFO







# Einsatz von Message Queues

- Consumer “pullt“
- Sprachunabhängig





# Einsatz von Message Queues

- Consumer “pullt“
- Sprachunabhängig
- Skalierbarkeit und Fallback





# Einsatz von Message Queues

- Beispiele:





# Einsatz von Message Queues

- Beispiele:
  - RabbitMQ
  - ActiveMQ





# RabbitMQ



# RabbitMQ

- Advanced Message Queuing Protocol (AMQP)

# RabbitMQ

- Advanced Message Queuing Protocol (AMQP)
  - Industrie Standard

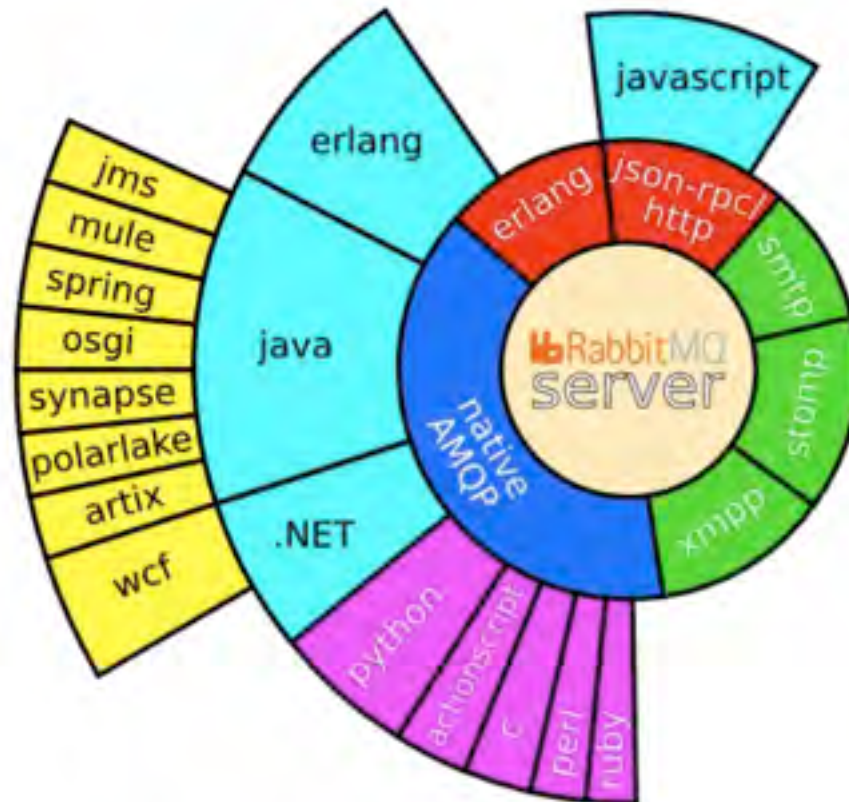
# RabbitMQ

- Advanced Message Queuing Protocol (AMQP)
  - Industrie Standard
- Sprachunabhängig
- OpenSource Client Lösungen





# RabbitMQ Client Übersicht

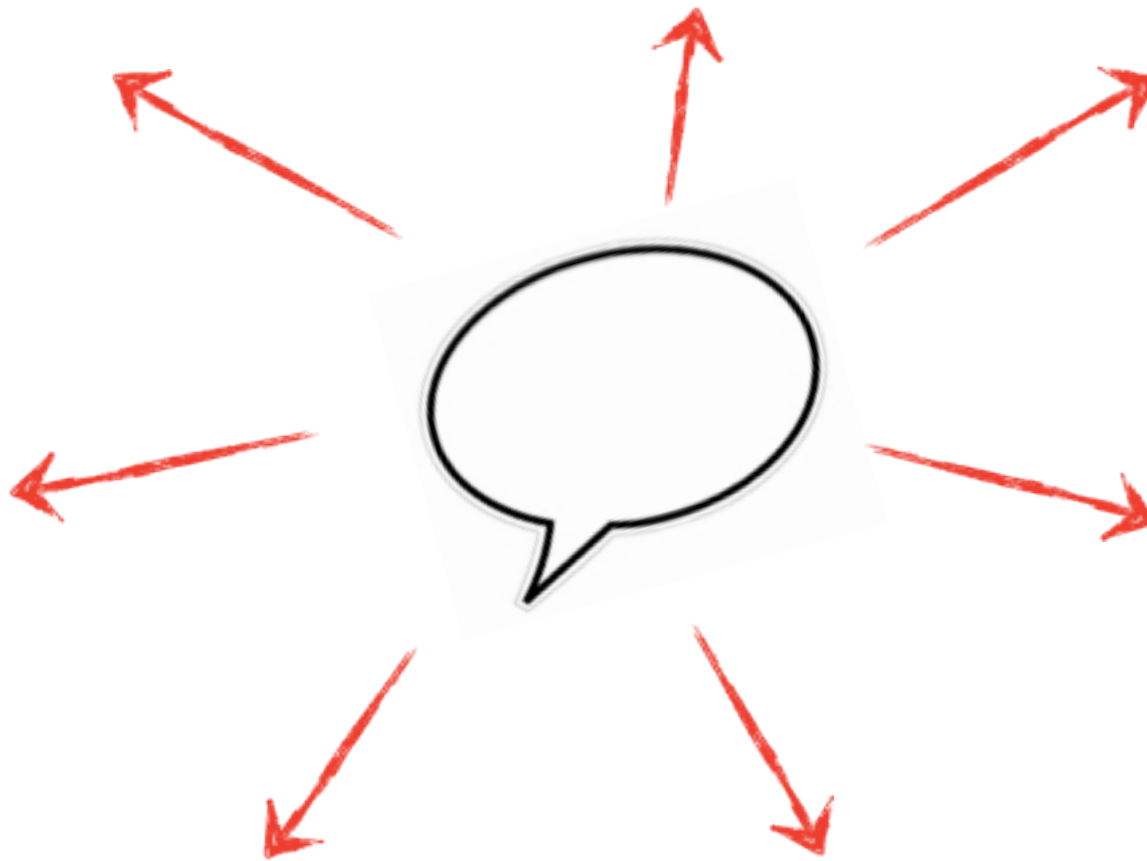


- Transport
- Gateway
- Client
- Adapter





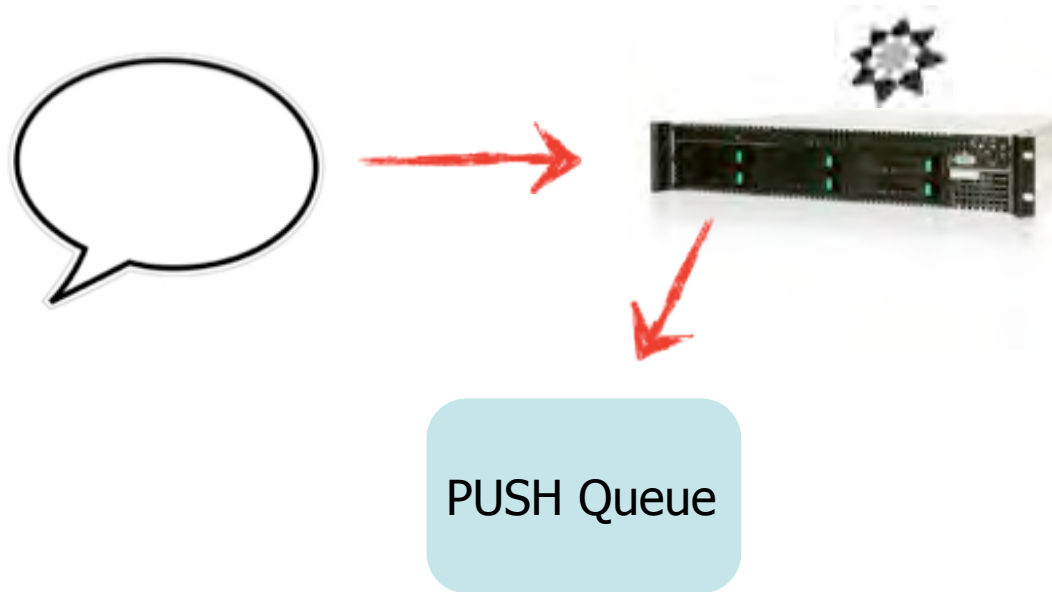
# Queues bei den VZ Netzwerken



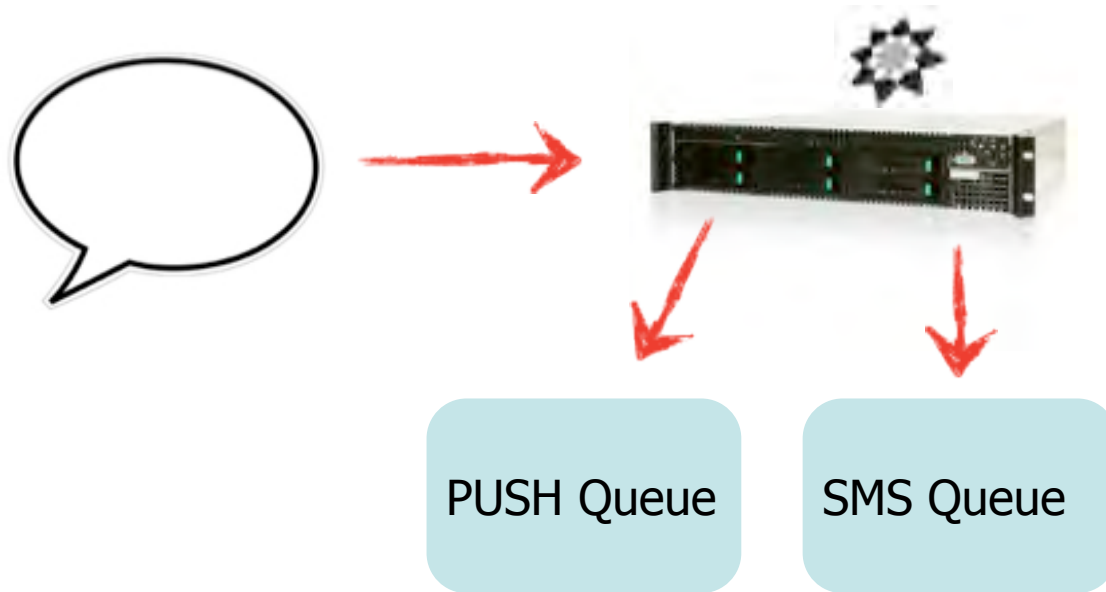
# Queues bei den VZ Netzwerken



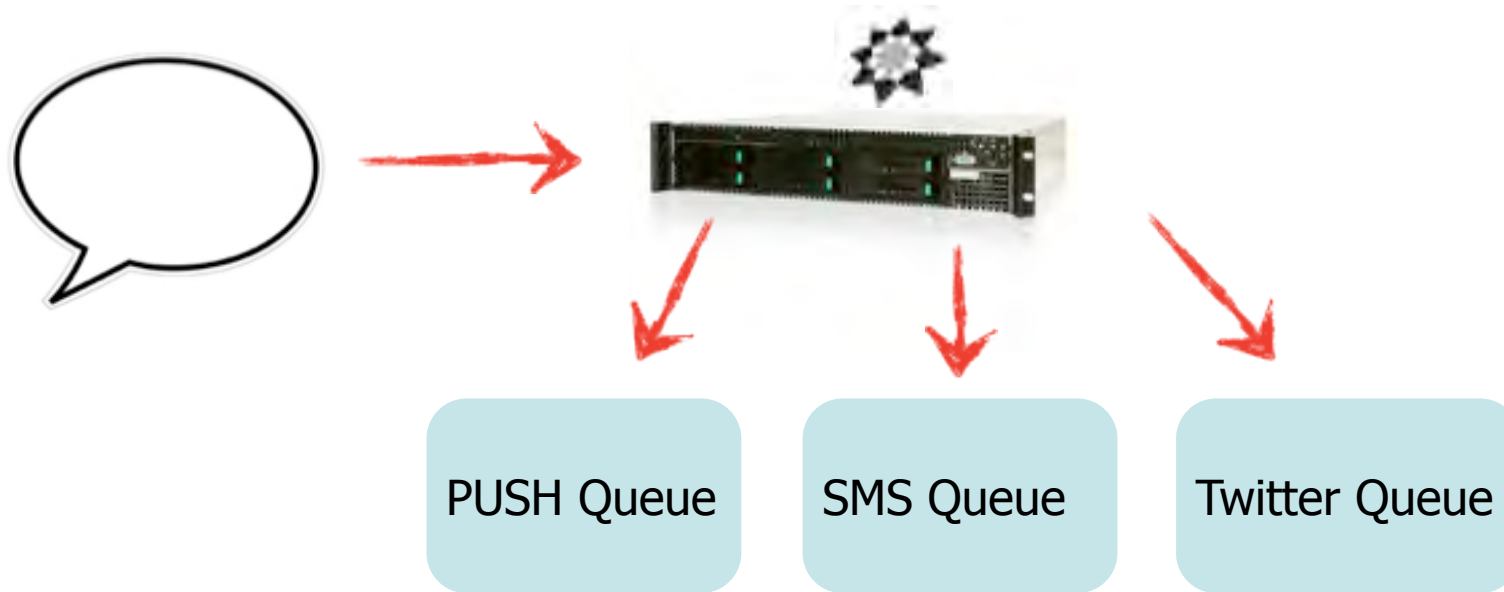
# Queues bei den VZ Netzwerken



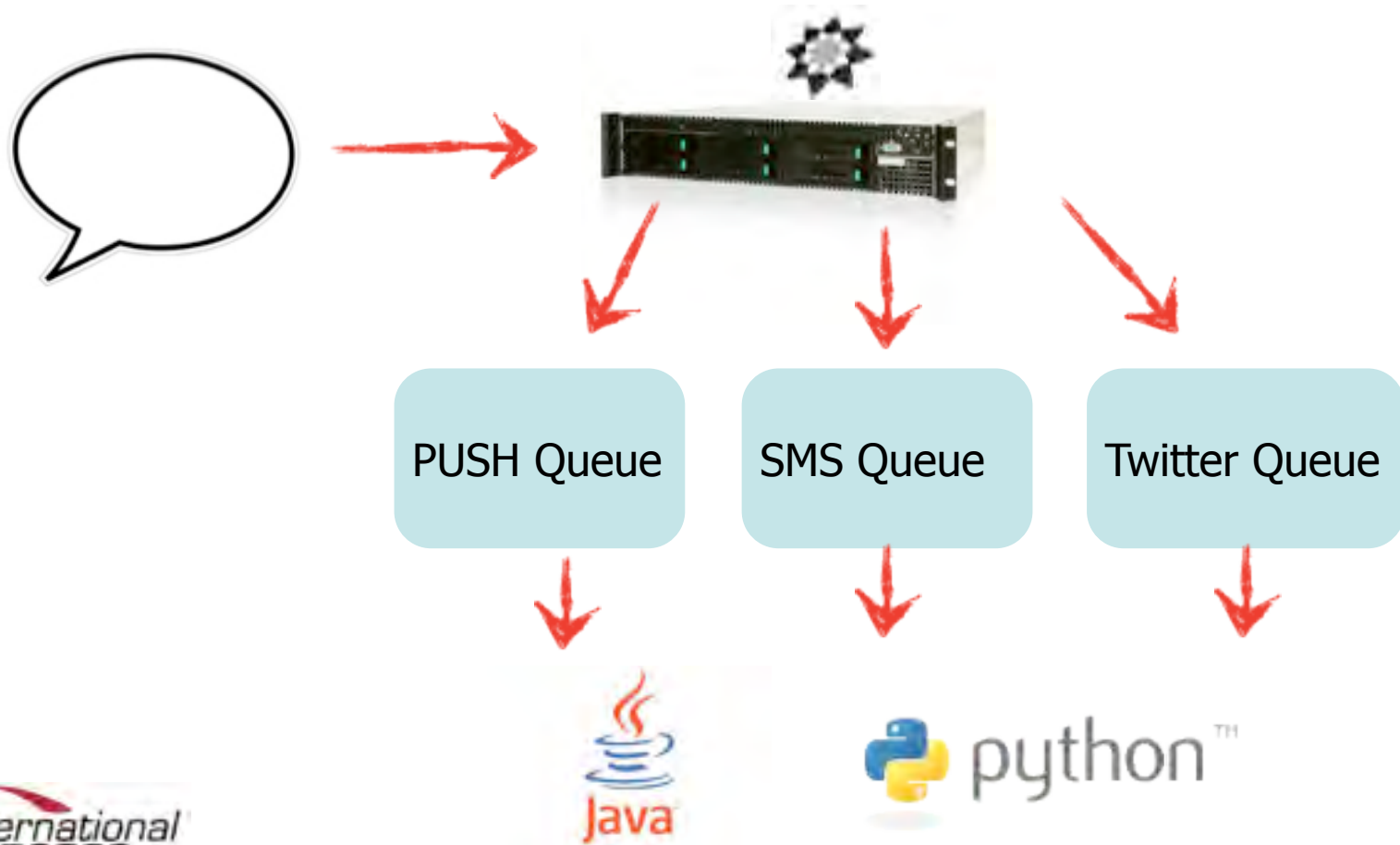
# Queues bei den VZ Netzwerken



# Queues bei den VZ Netzwerken

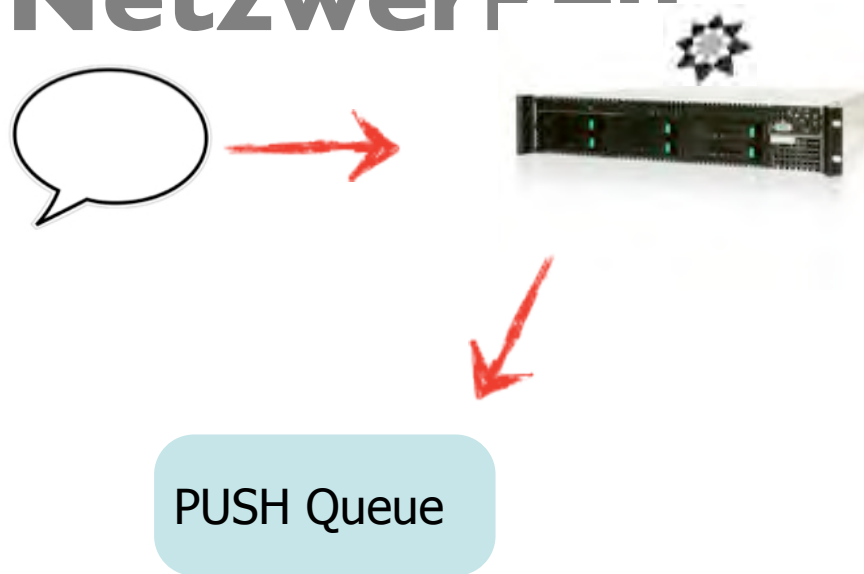


# Queues bei den VZ Netzwerken





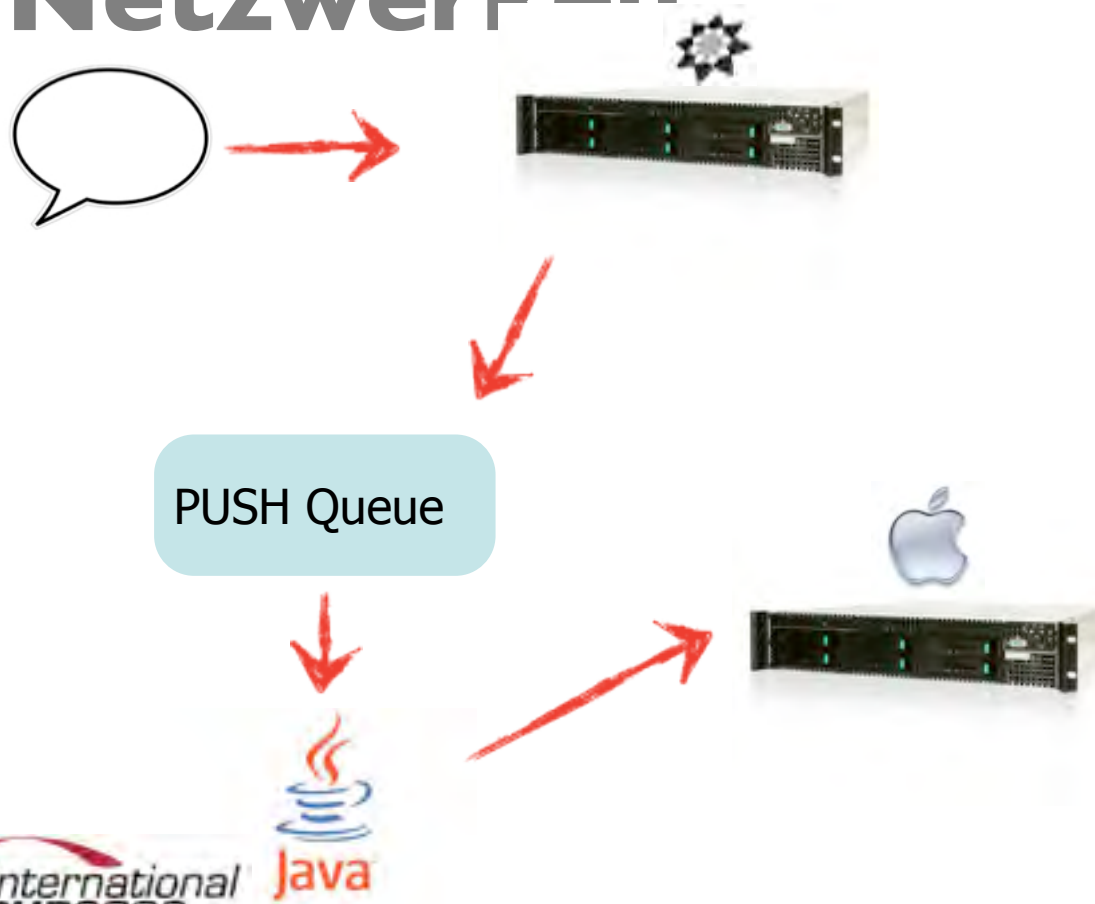
# Push-Ablauf bei den VZ Netzwerken





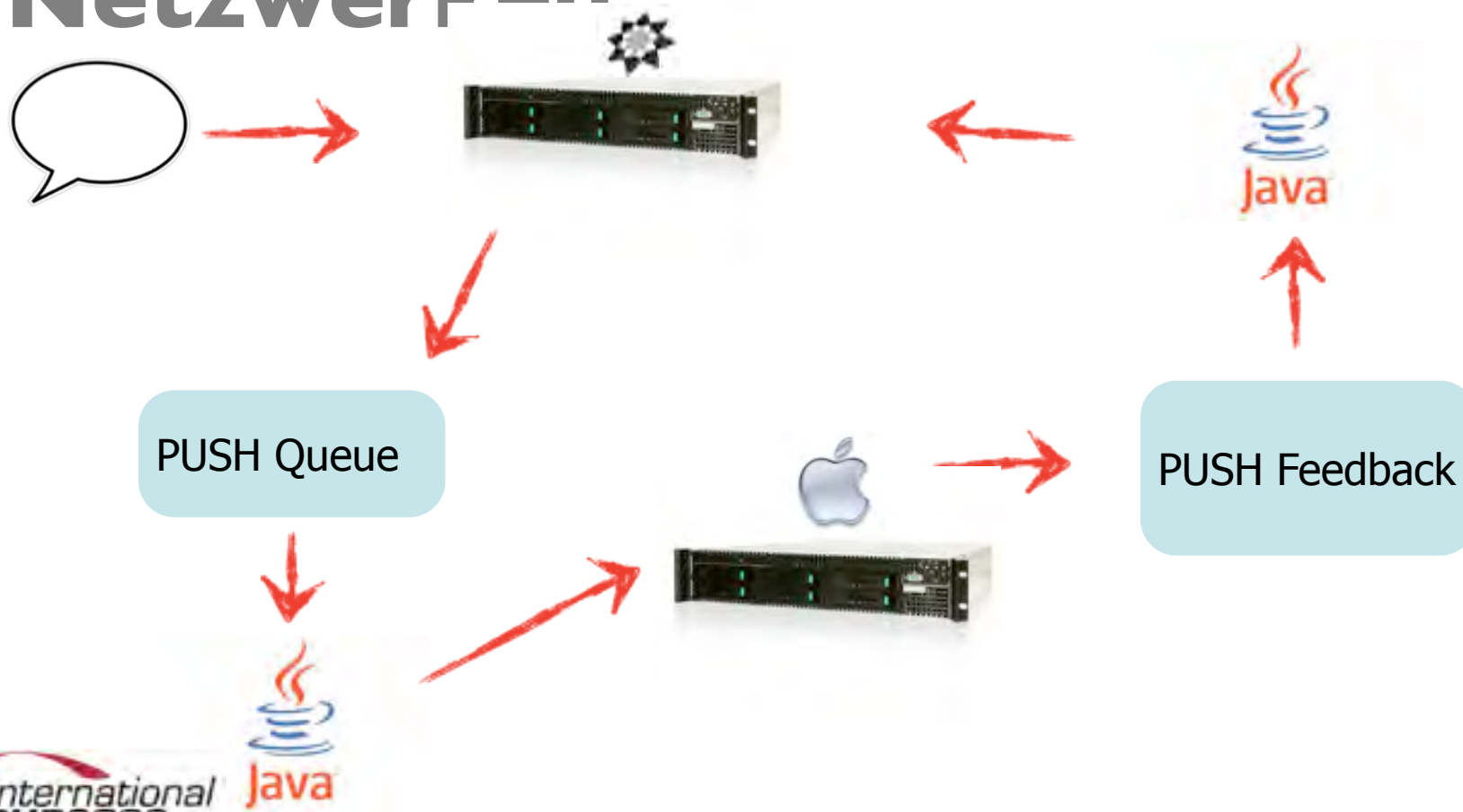


# Queuing Ablauf bei den VZ Netzwerken

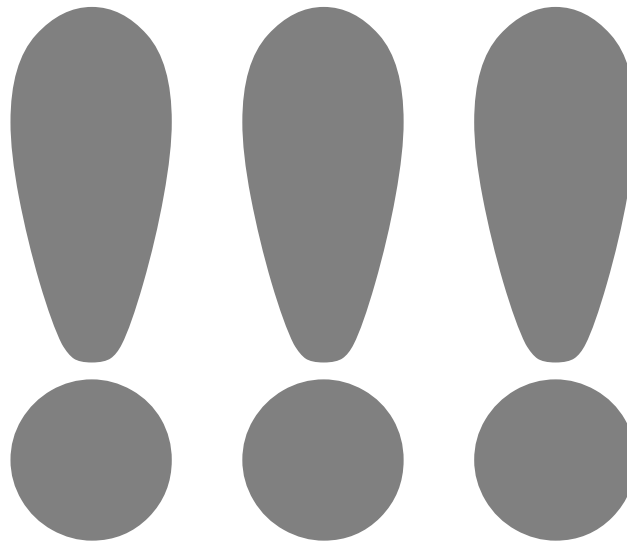




# Queuing Ablauf bei den VZ Netzwerken



# Lessons Learned



# Lessons Learned

- Clients sollten kaum Businesslogik enthalten
  - Webframework hält oft bereits notwendige Logik

# Lessons Learned

- Clients sollten kaum Businesslogik enthalten
  - Webframework hält oft bereits notwendige Logik
- Keinen zweiten Datenbanklayer schreiben
- Stattdessen API Requests an das Webframework stellen
- Sinnvolles, einheitliches Messageformat verwenden





# Push Beispiele



# Push Beispiel

# Publisher



# Push Beispiel: Publisher

- PHP C Extension:
  - <http://code.google.com/p/php-amqp>
- Veröffentlicht von den VZ Netzwerken



# Push Beispiel: Publisher

- PHP C Extension:
  - <http://code.google.com/p/php-amqp>
- Veröffentlicht von den VZ Netzwerken
- Wesentlich schneller als die reine Implementation in PHP

# Push Beispiel: Publisher

```
$msg = "hello webtech";
```



# Push Beispiel: Publisher

```
$msg = "hello webtech";
```

```
$connection = amqp_connection_popen("localhost", 5672);
```

# Push Beispiel: Publisher

```
$msg = "hello webtech";
```

```
$connection = amqp_connection_popen("localhost", 5672 );
```

```
amqp_login($connection, $user, $pass, $vhost);
```



# Push Beispiel: Publisher

```
$msg = "hello webtech";
```

```
$connection = amqp_connection_popen("localhost", 5672 );
```

```
amqp_login($connection, $user, $pass, $vhost);
```

```
amqp_basic_publish($connection, ..., $msg, ..., $options);
```

# Push Beispiel

# Consumer





# Push Beispiel: Consumer

- RabbitMQ Java Lib
  - <http://www.rabbitmq.com/java-client.html>





# Push Beispiel: Consumer

```
Consumer consumer = new QueueingConsumer(...);
```







# Push Beispiel: Consumer

```
Consumer consumer = new QueueingConsumer(...);
```

```
while (true) {
```

```
    // Get delivery-item ...
```

```
    QueueingConsumer.Delivery item = consumer.nextDelivery();
```

```
    // Do something with delivery ...
```

```
    ...
```

```
}
```



# Push Beispiel

# Java APNS



# Push Beispiel: Java APNS

- Apple Push Notification Service Provider for Java
  - <http://code.google.com/p/javapns/>





# Java Push Beispiel - APNS

```
manager = PushNotificationManager.getInstance();
```





# Java Push Beispiel - APNS

```
manager = PushNotificationManager.getInstance();
```

```
client = manager.getDevice("iPhone");
```





# Java Push Beispiel - APNS

```
manager = PushNotificationManager.getInstance();
```

```
client = manager.getDevice("iPhone");
```

```
simplePayload = new Payload();  
simplePayload.addAlert(message);  
simplePayload.addBadge(msgCount);
```





# Java Push Beispiel - APNS

```
manager = PushNotificationManager.getInstance();
```

```
client = manager.getDevice("iPhone");
```

```
simplePayload = new Payload();  
simplePayload.addAlert(message);  
simplePayload.addBadge(msgCount);
```

```
manager.initializeConnection(host, port, ssl, pwd, ssl_type, device);  
manager.sendNotification(client, simplePayload);
```



# Fragen?



# Danke fürs Zuhören!

[developer.studivz.net](http://developer.studivz.net)

[max@vz.net](mailto:max@vz.net)

[twitter.com/MaxHorvath](https://twitter.com/MaxHorvath)

[steffen@vz.net](mailto:steffen@vz.net)

[twitter.com/sirrgang](https://twitter.com/sirrgang)

