xmpp, python..... and fluffy clouds

mendix
NO CODE, JUST GLORY.

Mendix?

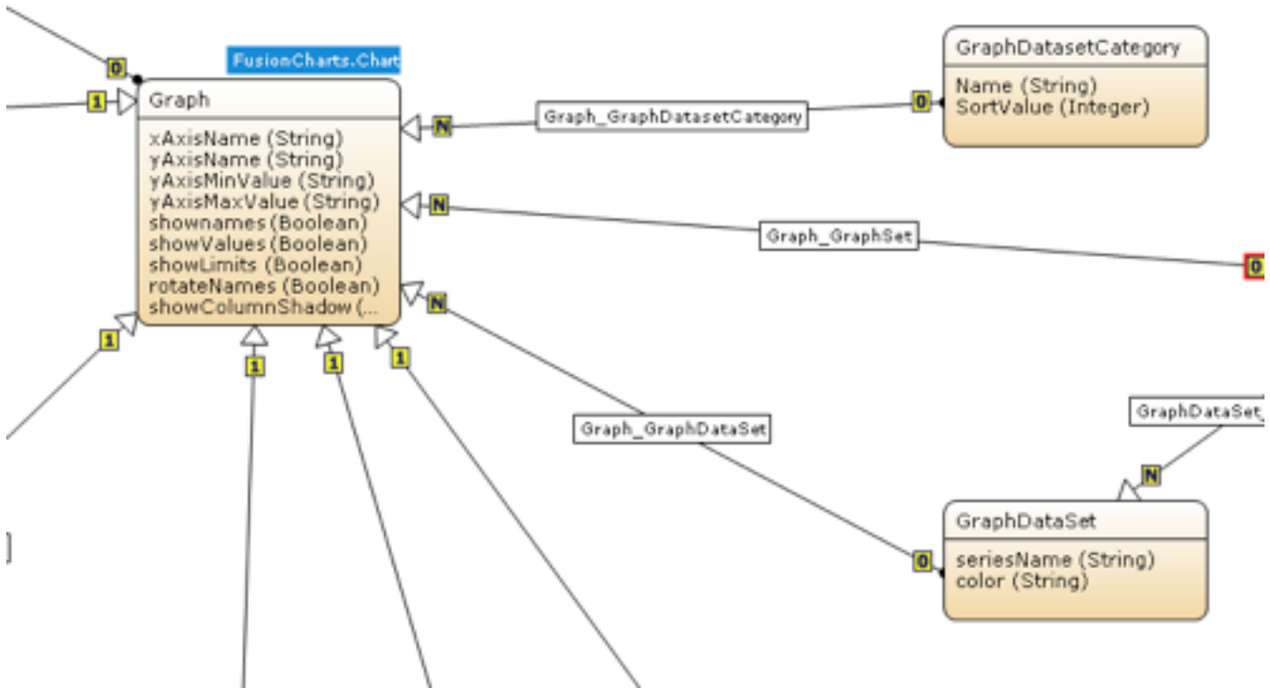
django models.py:

```python
# encoding:utf-8
from django.db import models

class Tag(models.Model):
    name       = models.CharField(max_length=255, unique=True)
    created_by = models.ForeignKey(User, related_name='created_tags')
    # Denormalised data
    used_count = models.PositiveIntegerField(default=0)

    objects = TagManager()

    class Meta:
        db_table = u'tag'
        ordering = ('-used_count', 'name')

    def __unicode__(self):
        return self.name

class Comment(models.Model):
    content_type   = models.ForeignKey(ContentType)
    object_id      = models.PositiveIntegerField()
    content_object = generic.GenericForeignKey('content_type', 'object_id')
    user           = models.ForeignKey(User, related_name='comments')
    comment        = models.CharField(max_length=300)
    added_at       = models.DateTimeField(default=datetime.datetime.now)

    class Meta:
        ordering = ('-added_at',)
        db_table = u'comment'
    def __unicode__(self):
        return self.comment
```

django models.py:

```python
# encoding:utf-8
from django.db import models

class Tag(models.Model):
    name       = models.CharField(max_length=255, unique=True)
    created_by = models.ForeignKey(User, related_name='created_tags')
    # Denormalised data
    used_count = models.PositiveIntegerField(default=0)

    objects = TagManager()

    class Meta:
        db_table = u'tag'
        ordering = ('-used_count', 'name')

    def __unicode__(self):
        return self.name

class Comment(models.Model):
    content_type   = models.ForeignKey(ContentType)
    object_id      = models.PositiveIntegerField()
    content_object = generic.GenericForeignKey('content_type', 'object_id')
    user           = models.ForeignKey(User, related_name='comments')
    comment        = models.CharField(max_length=300)
    added_at       = models.DateTimeField(default=datetime.datetime.now)

    class Meta:
        ordering = ('-added_at',)
        db_table = u'comment'
    def __unicode__(self):
        return self.comment
```

mendix model:

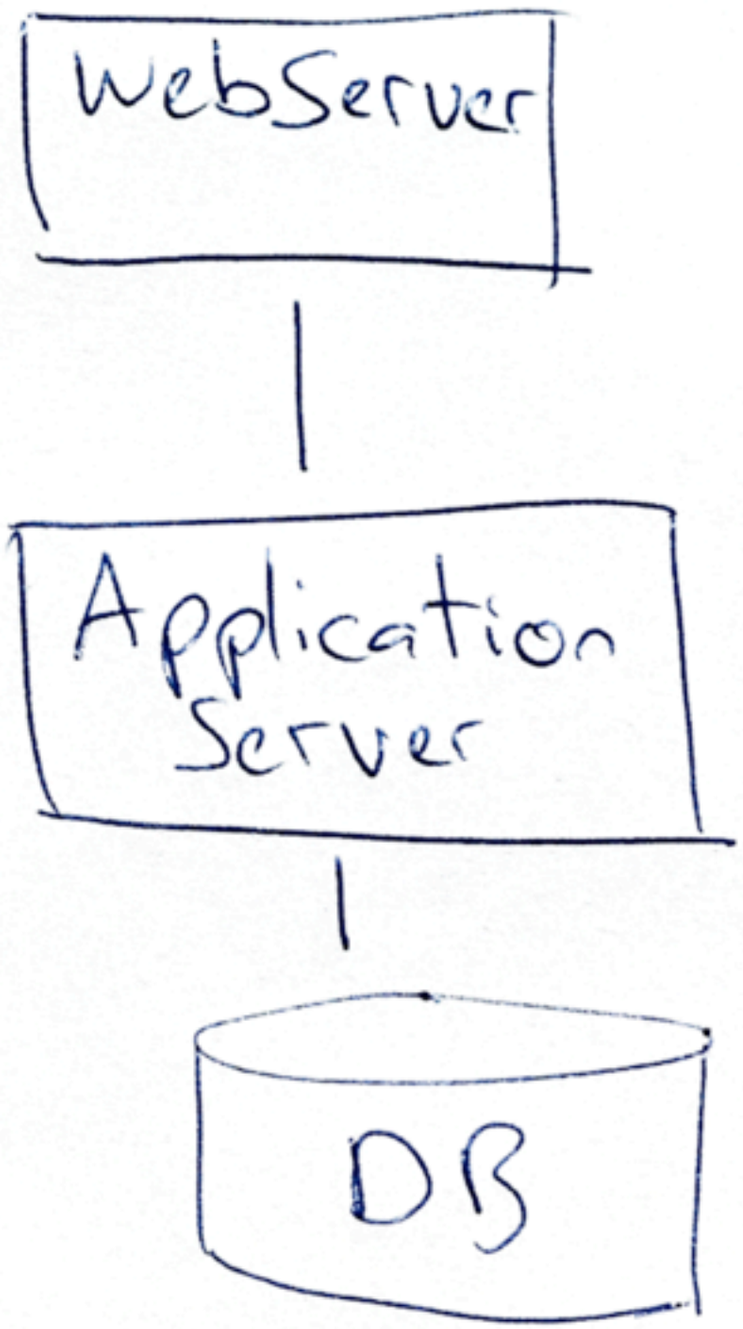


how do I ‘run’ this model?

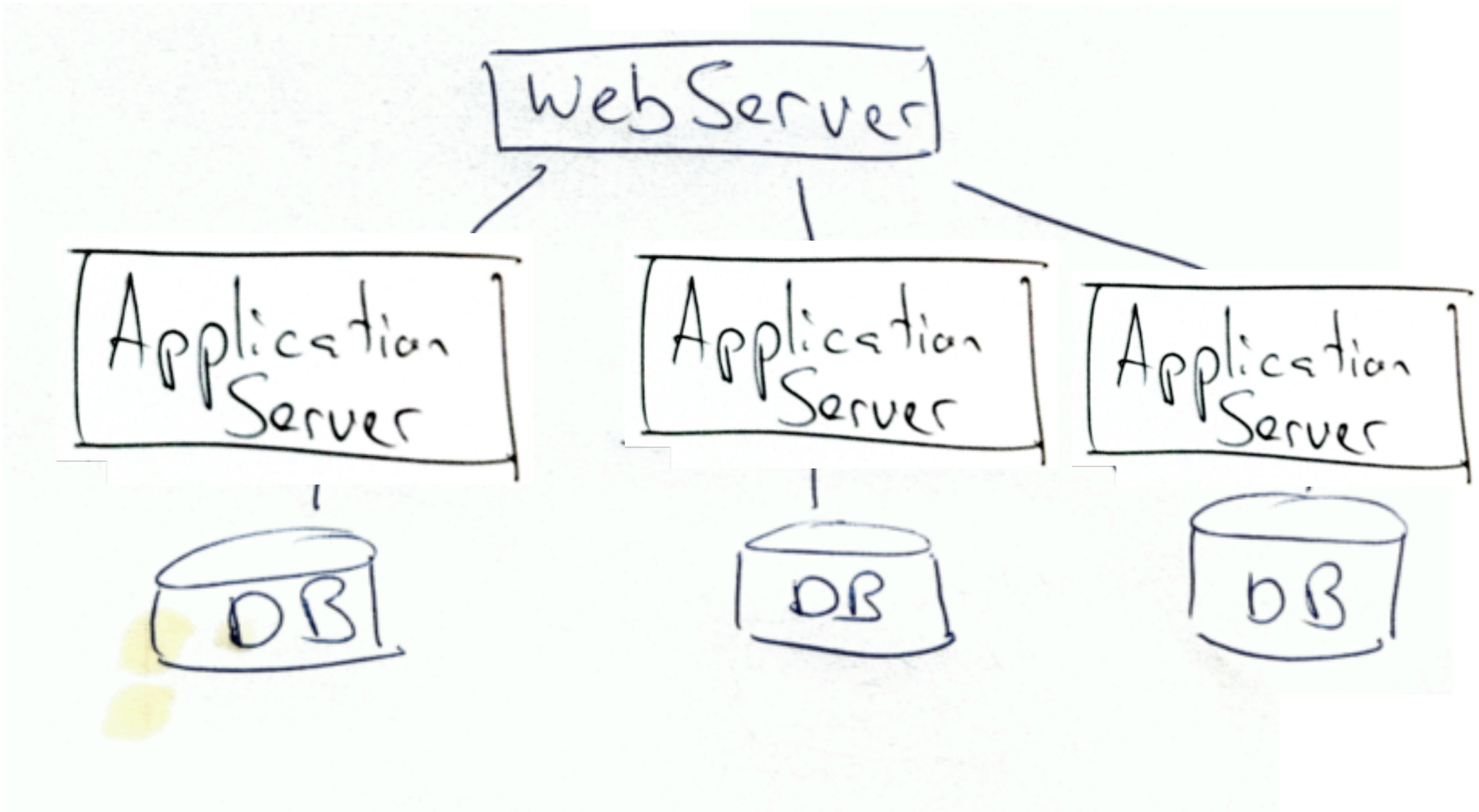
Saturday, June 25, 2011

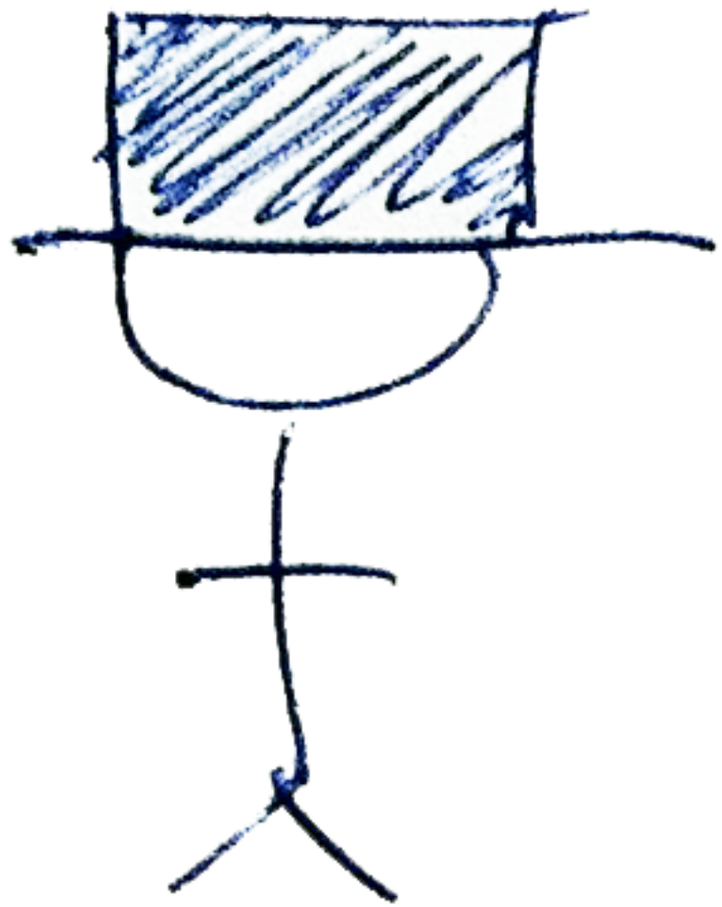

RTFM....

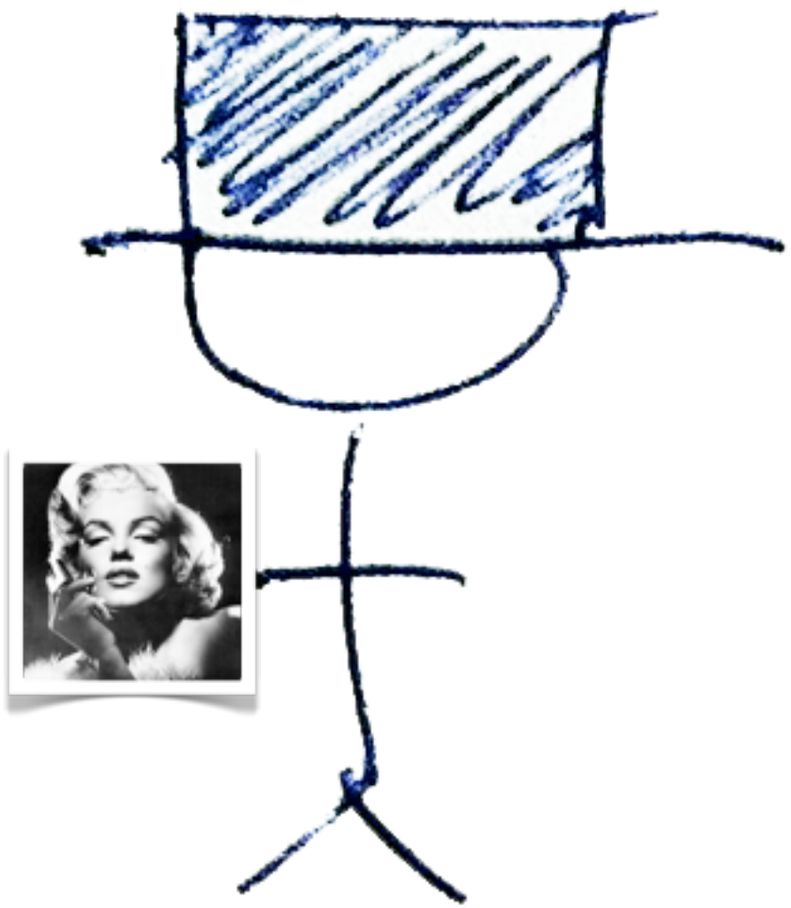
traditional three tier architecture



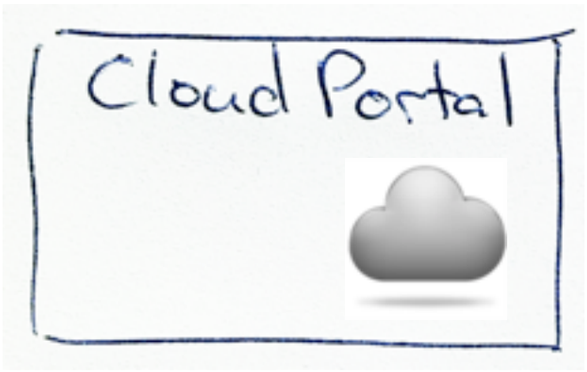
cloud architecture



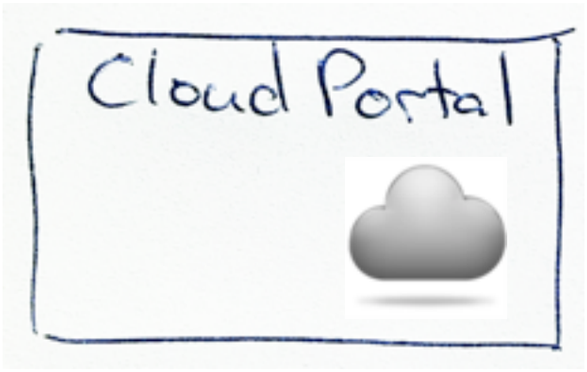


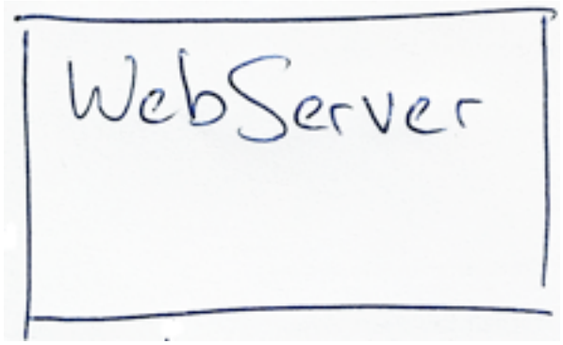


cloud-portal.mendix.com

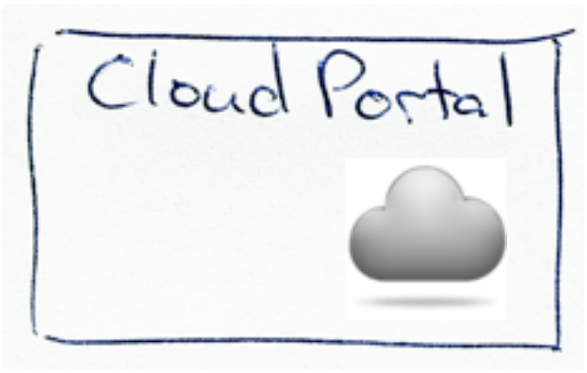

Cloud Portal

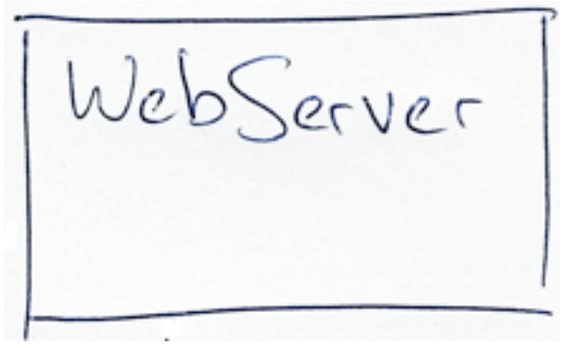

Application
Server


Cloud Portal

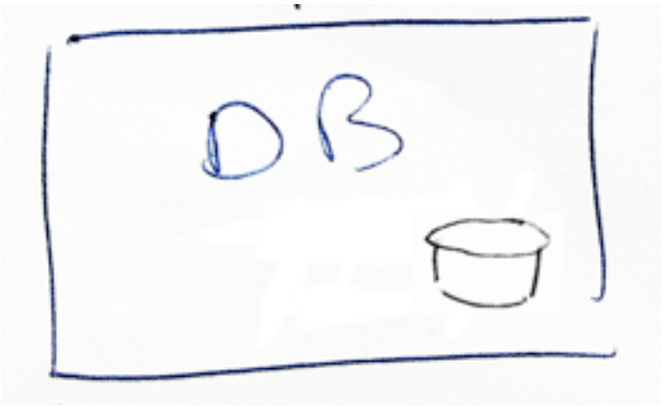

WebServer

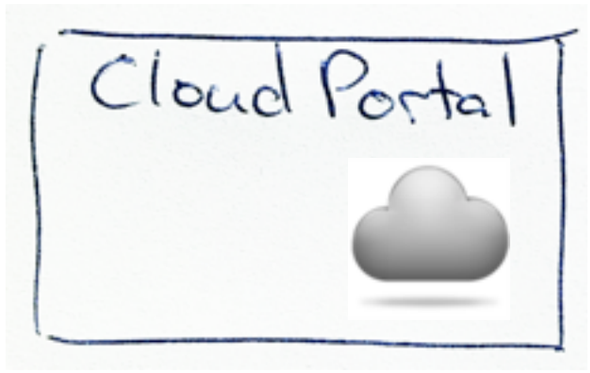

Application
Server

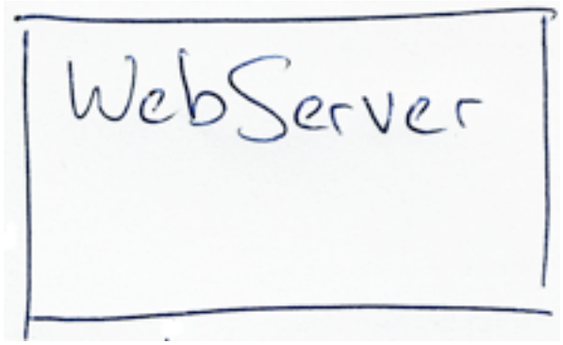

Cloud Portal


WebServer

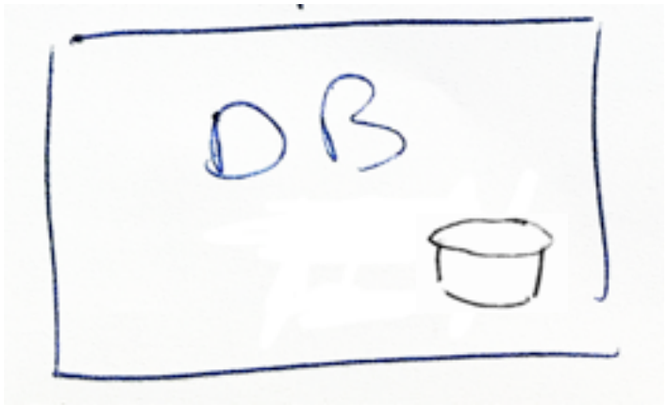

Application
Server


DB


Cloud Portal

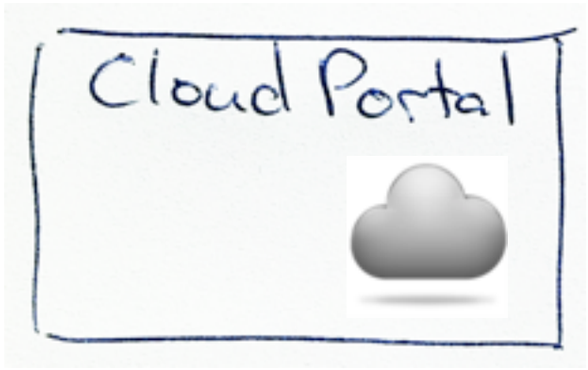

WebServer


Application
Server


DB

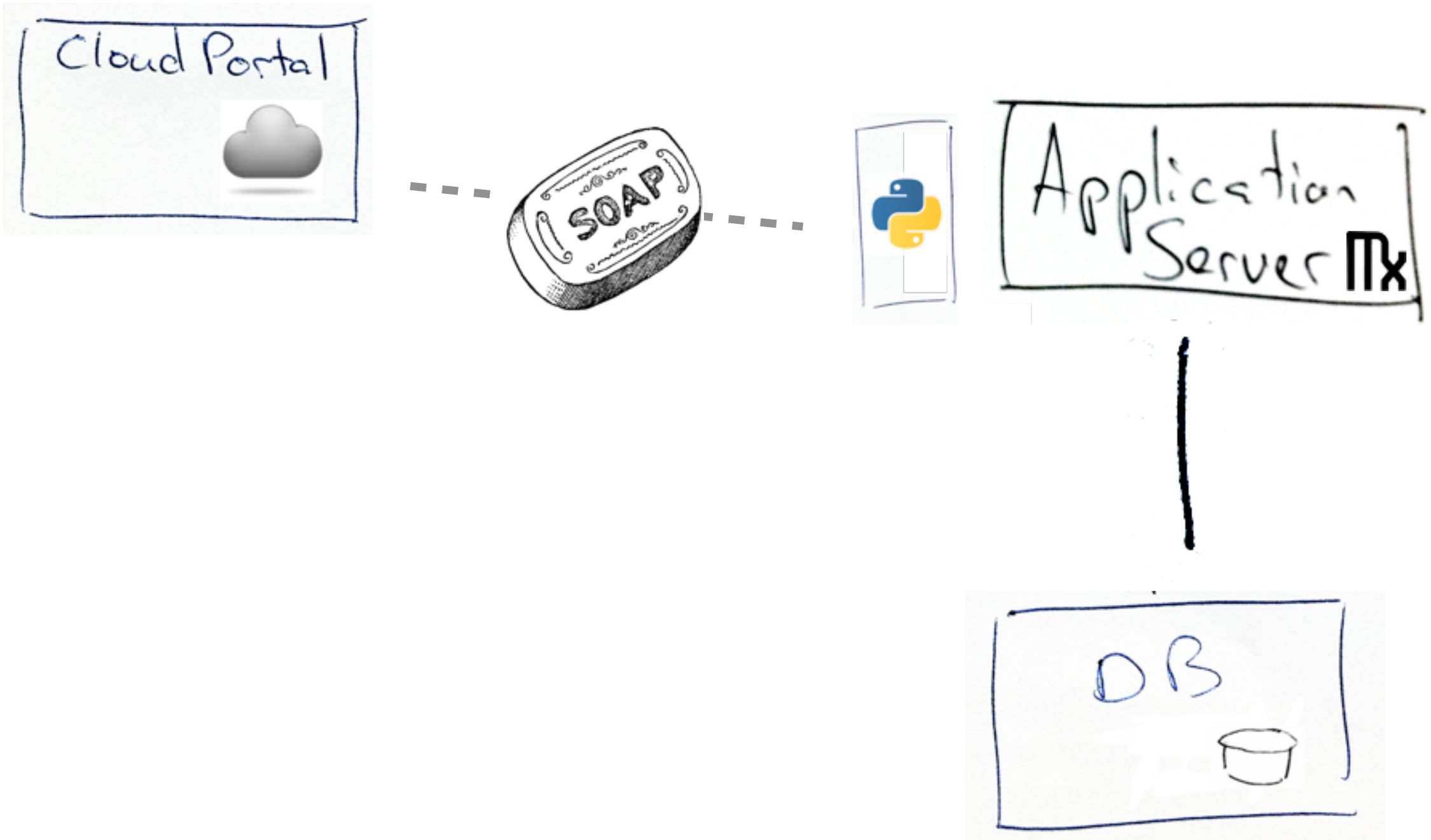
webserver configuration

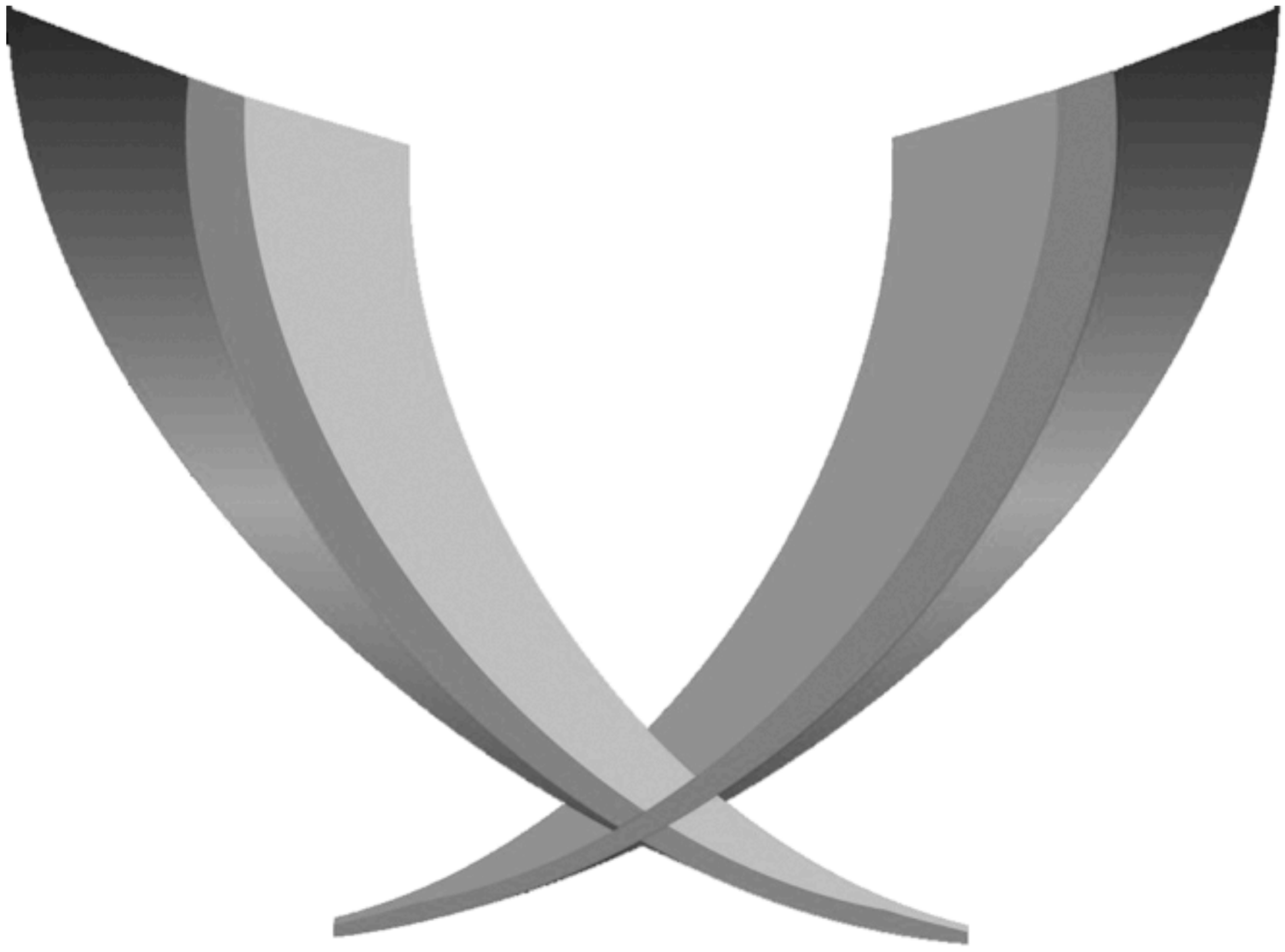
webserver configuration


Cloud Portal


Cloud Portal
SOAP


Cloud Portal
SOAP
Application Server
Mx


Cloud Portal
SOAP
Application Server
DB



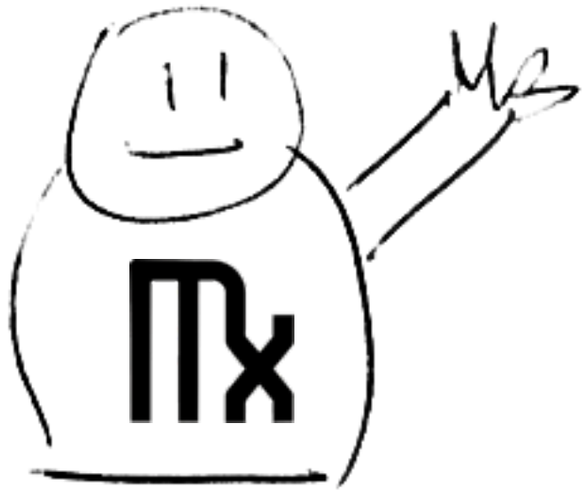

WebServer

please update the config

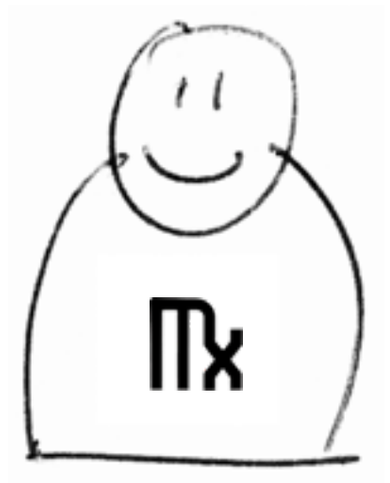
i need somewhere to put this model...

let me spawn that for you...

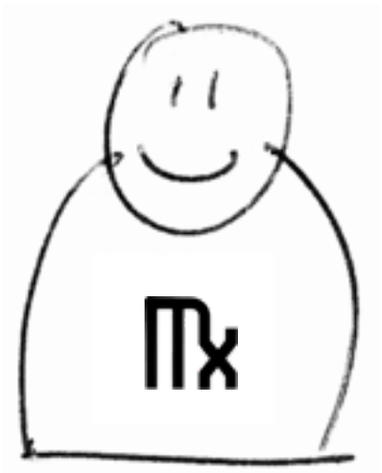
hello! i’m online



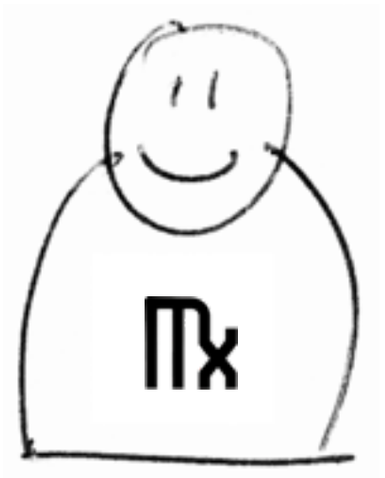

WebServer

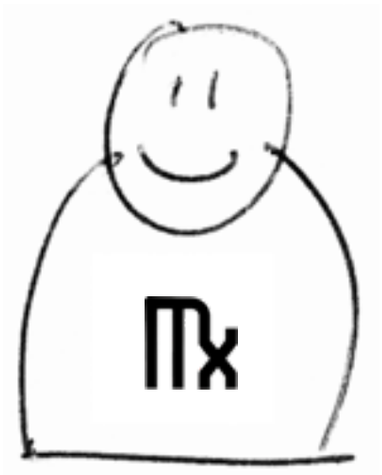

mx

i can haz db?

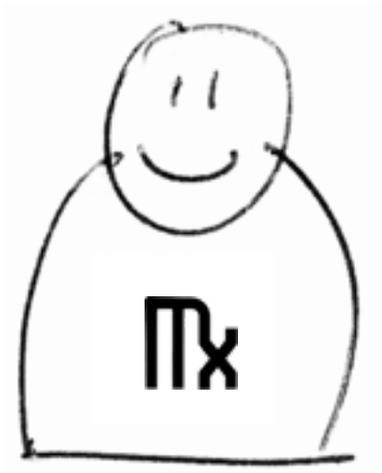


no db here



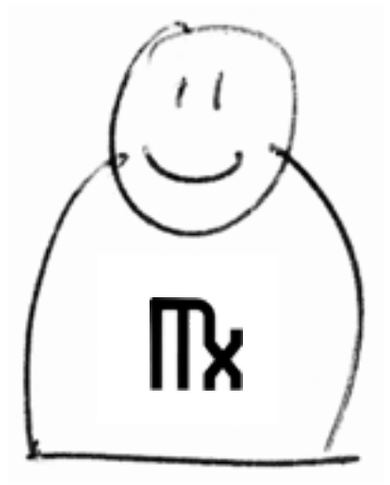


yes, i haz db!

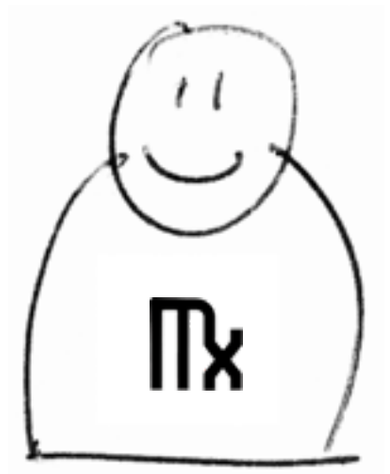


yay!

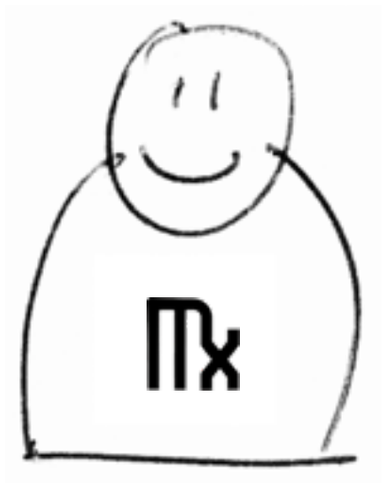

WebServer


mx


WebServer


mx


WebServer


Mx

but?

what have we learned?

easy to add processes

presence is cool

authorization & authentication out of the box

i can haz service?

feels nice

```
(\  /)
(~.~)
(>  <)
```
bunny thanks you for your time.

github.com/Achiel
github.com/Mendix
mendix.com

btw, we’re hiring!