



Stichting NIOC

Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden-nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

Go Technology

where it fits our problems



E. Micklei
Software Architect

bol.com 

bomvol winkels, bomvol voordeel

Building a platform

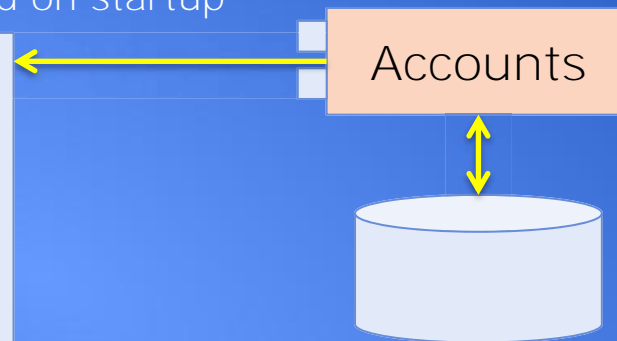
- developers needs abstraction from infrastructure
- so developers become agile
- self serviced
- shared, scalable building blocks



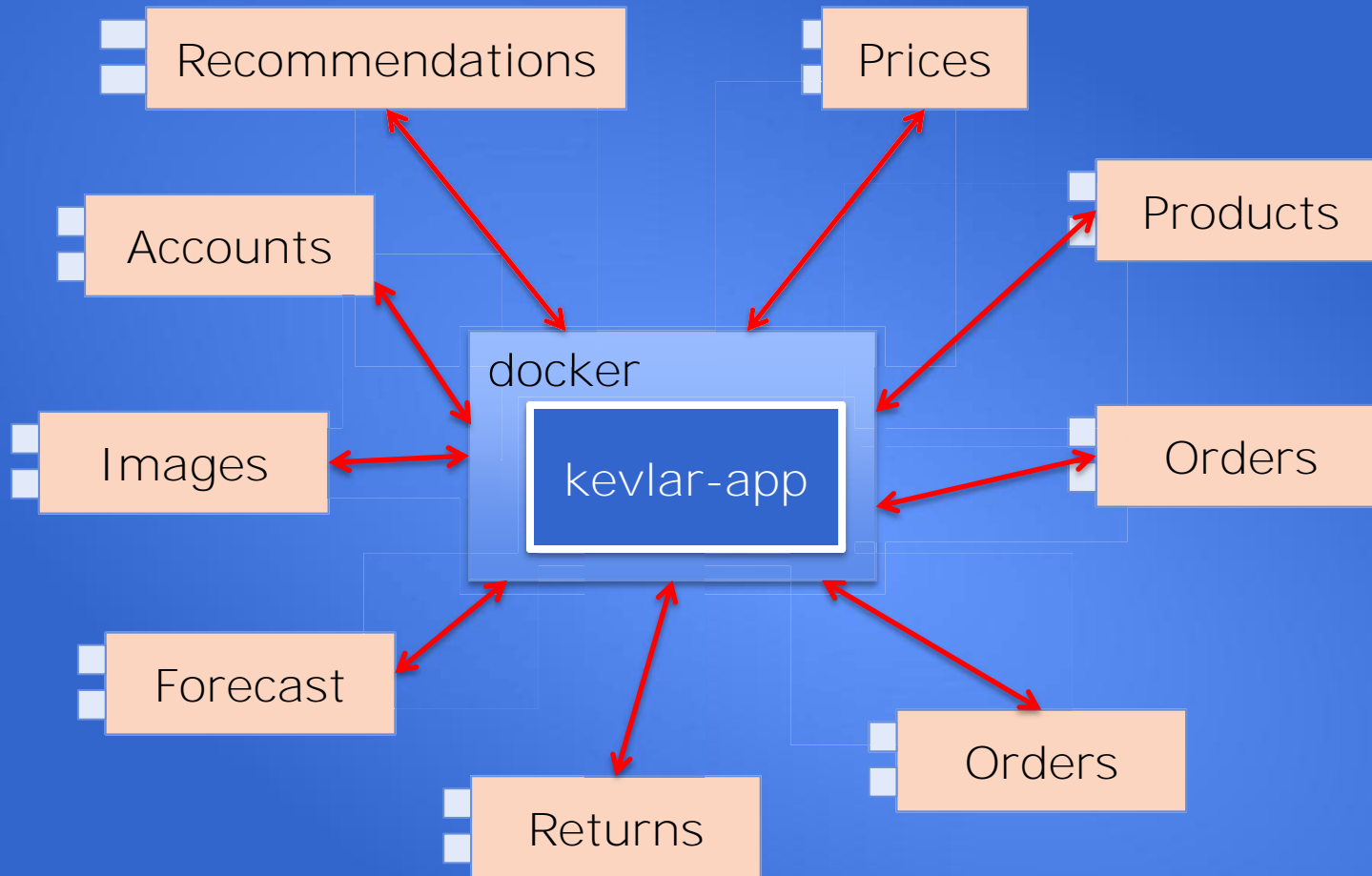
Application Configuration

```
db.host = test.db.001
db.port = ...
db.user = ...
db.max.pool = ...
db.pool.idle = ...
rest.port = 8080
logging.host = test.logstash
logging.port = ...
metrics.host = test.grafite
metrics.port = ...
```

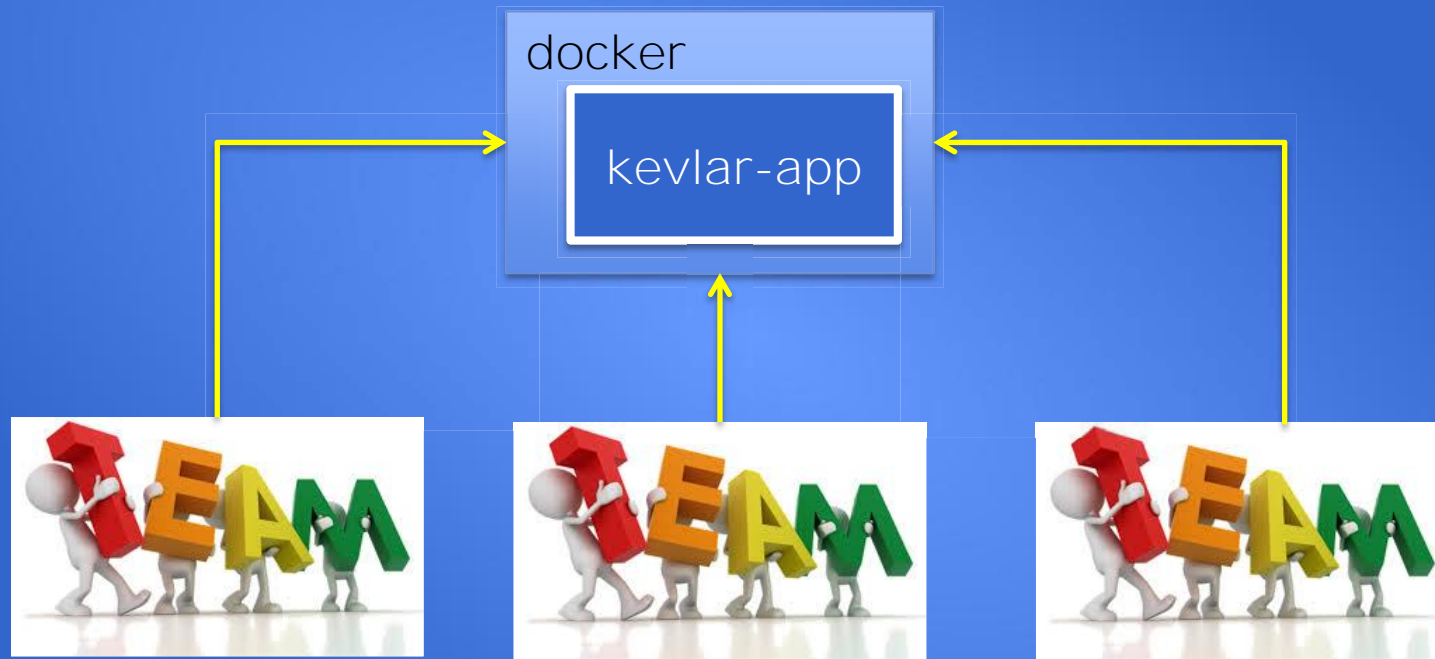
read on startup



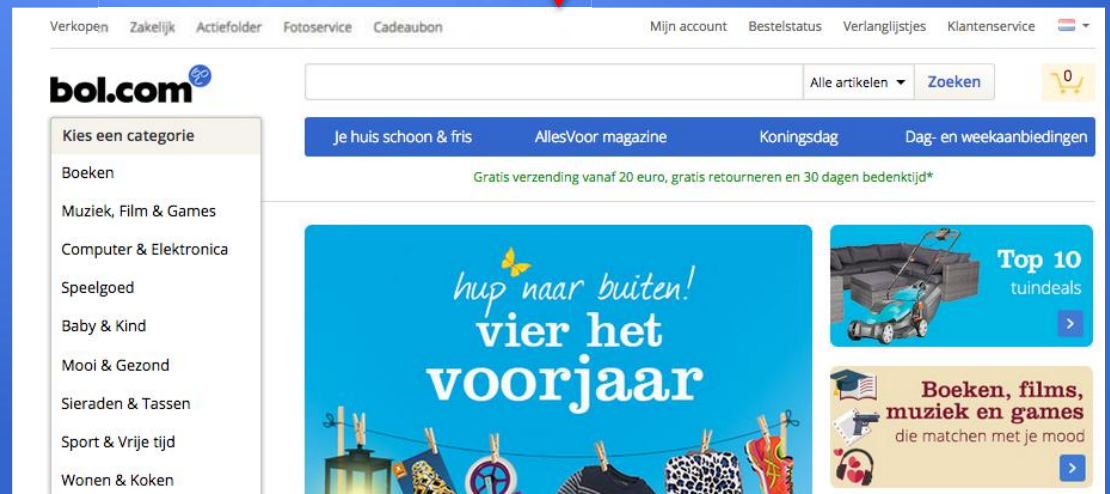
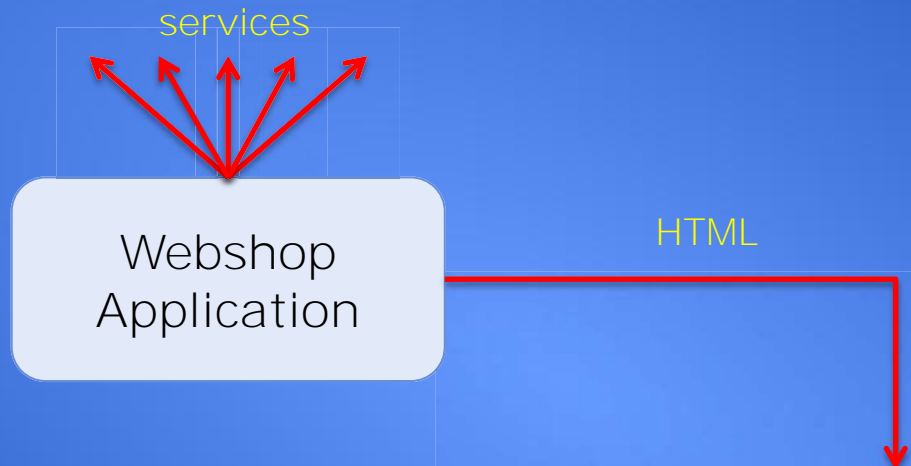
Configuration Service



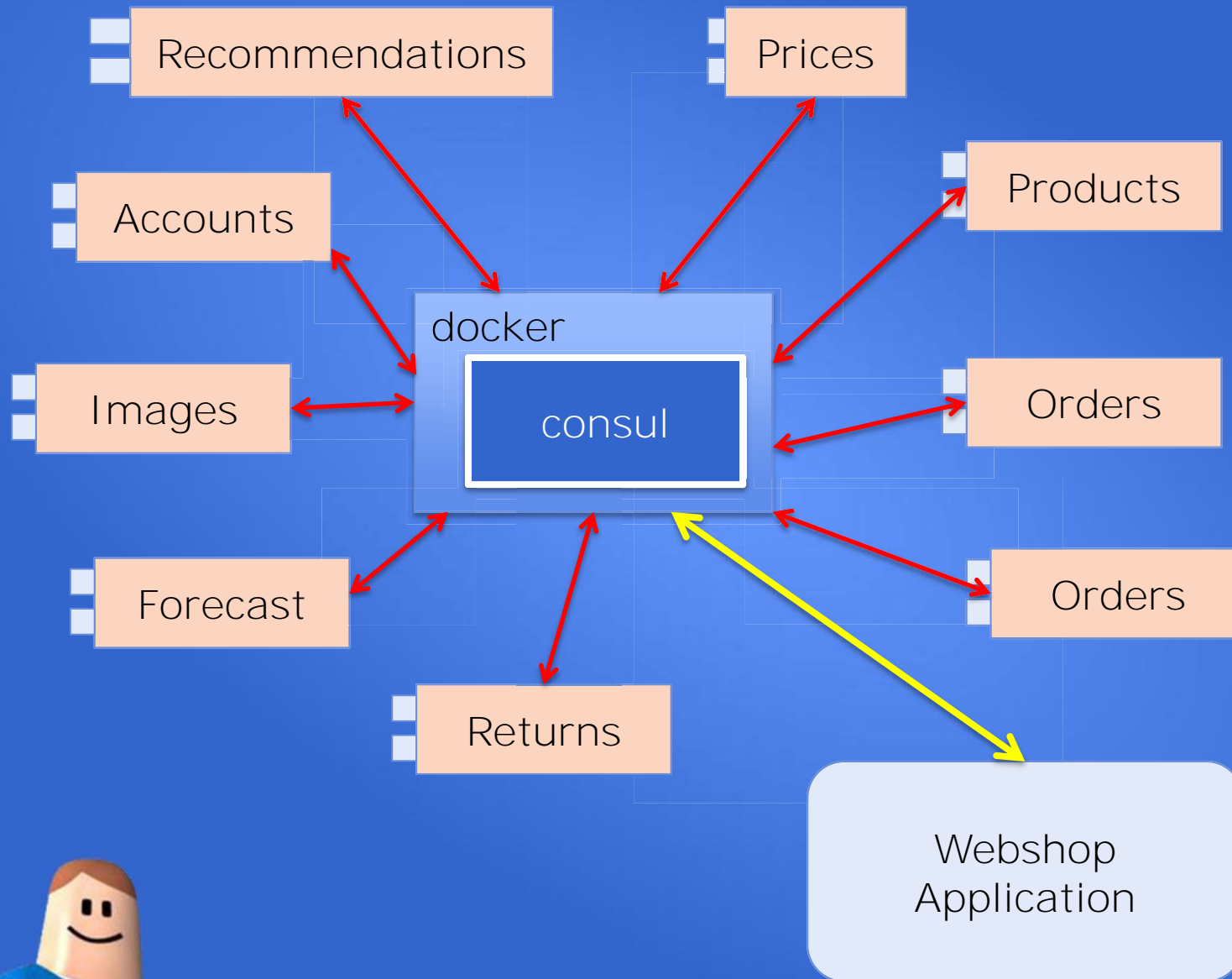
Configuration Mgmt.



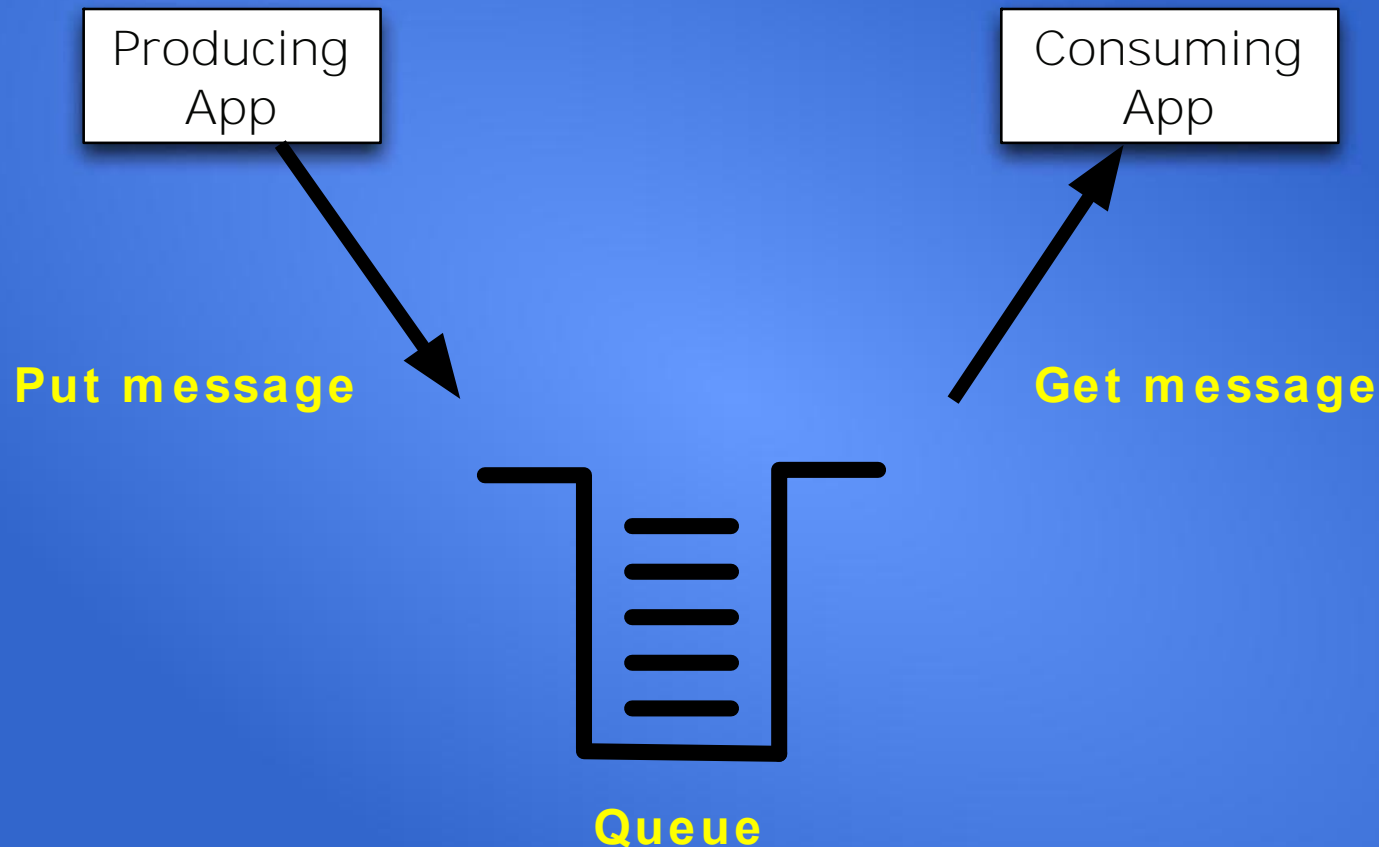
Service Discovery

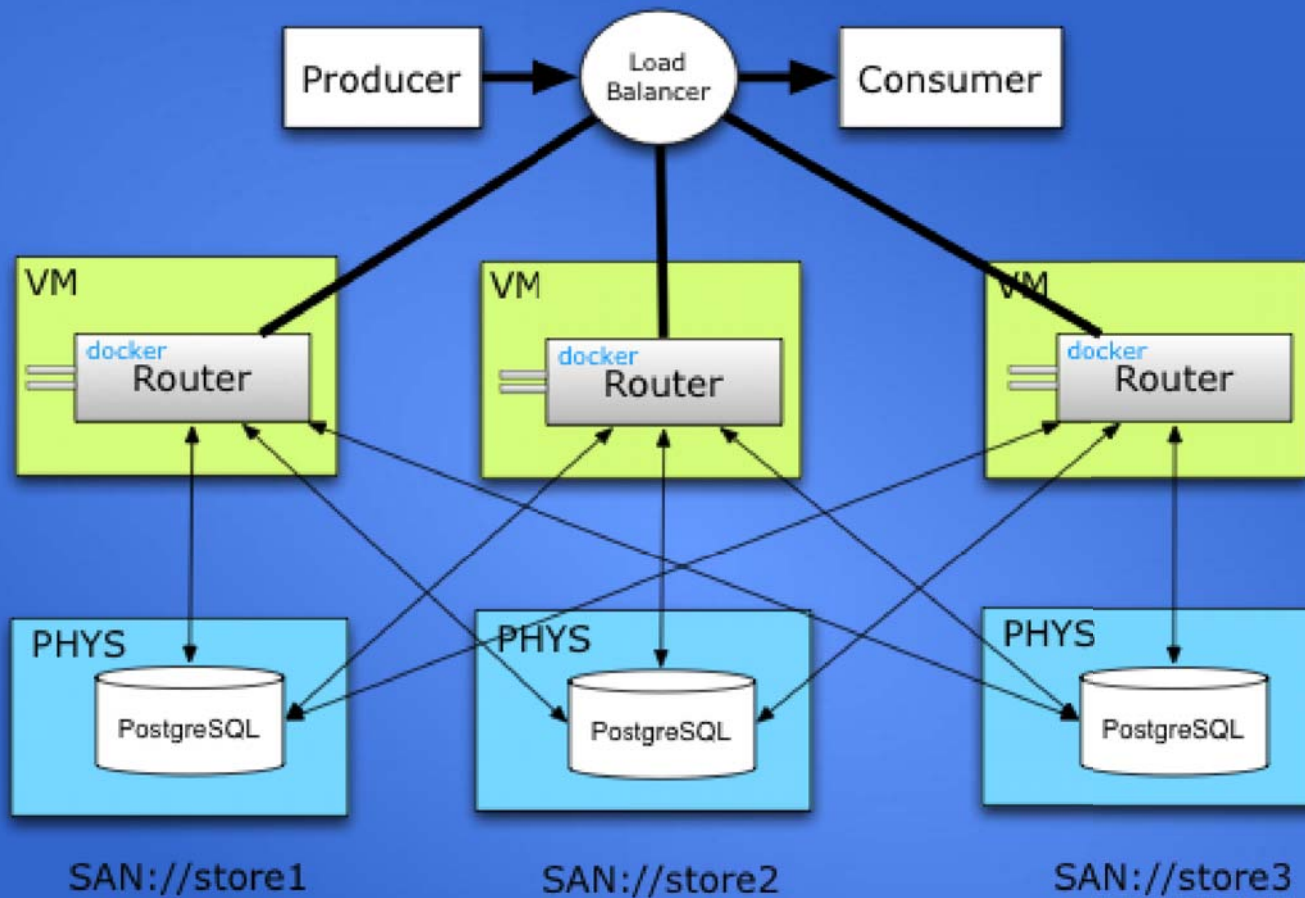


Service Discovery



Scaleable Messaging





BoQS - Bol.com Queueing System



REST api BoQS

```
func RegisterQueueResource(q QueueResource, m MessageResource, container *restful.Container) {  
    ws := new(restful.WebService)  
    ws.  
        ApiVersion("1.0").  
        Path("/v1/queues").  
        Doc("Consuming messages from queues.").  
        Consumes(restful.MIME_JSON, restful.MIME_XML).  
        Produces(restful.MIME_JSON, restful.MIME_XML)  
  
    nameParam := ws.PathParameter("name", "name of the queue").DataType("string")  
  
    ws.Route(ws.GET("/").To(q.getQueues).Operation("getQueues").  
        Doc("Return all queues").  
        Do(Returns200, Returns500))  
  
    ws.Route(ws.GET("/{name}/messages/count").To(q.getQueueSize).  
        // docs  
        Doc("Number of messages stored in the queue.").  
        Operation("getQueueSize").  
        Param(nameParam).  
        Do(Returns200, Returns500))  
}
```



Database access Kevlar

```
func (d PgPropertyDao) GetProperties(query PropertyAccess) (*properties.Properties, error) {
    props := properties.NewProperties()

    err := doWithConnection(d.pool, MaxConnectionRetries(), func(conn *pgx.Conn) error {
        rows, err := conn.Query("select * from get_properties($1,$2,$3);", query.Label, query.Version, query.Query)
        if err != nil {
            glog.Errorf("GetProperties -> unable to get_properties(%#v):%v", query, err)
            return err
        }
        defer rows.Close()

        for rows.Next() {
            var key_name string
            var key_description string
            var key_is_secret bool
            var key_deleted_at []byte
            var key_value string
            var key_type string
            rows.Scan(&key_name, &key_description, &key_is_secret, &key_deleted_at, &key_value, &key_type)
            if rows.Err() != nil {
                glog.Errorf("GetProperties -> rows.Scan failed:%v", rows.Err())
                return rows.Err()
            }
            props.SetComment(key_name, key_description)
            props.Set(key_name, key_value)
        }
        return nil
    })
    return props, err
}
```



Using Go in our "cloud"

- easy to learn
- no runtime dependencies
- rich standard library
- lot of innovations out there



Using Go in our "cloud"

- high quality third party libs
- developer friendly
 - simple syntax
 - auto formatting
 - good documentation





Let's Go

