OBJECT COMPUTING
HOME TO MICRONAUT

WEBINAR

MICRONAUT MULTITENANCY

SERGIO DEL AMO

SERGIO DEL AMO

- MICRONAUT / GRAILS OCI TEAM
- GUADALAJARA, SPAIN
- CURATOR OF GROOVYCALAMARI.COM
- PODCAST HOST OF PODCAST.GROOVYCALAMARI.COM
- GREACH Conference organizer
- @SDELAMO
- HTTP://SERGIODELAMO.ES

Multi-Tenancy, as it relates to software development, is when a single instance of a an application is used to service multiple clients (tenants) in a way that each tenants' data is isolated from the other.

INSTALLATION

MULTI-TENANCY INSTALLATION

build.gradle

```
dependencies {
    ...
    ..
    .
    compile: 'io.micronaut:micronaut-multitenancy'
}
```

OBJECT COMPUTING

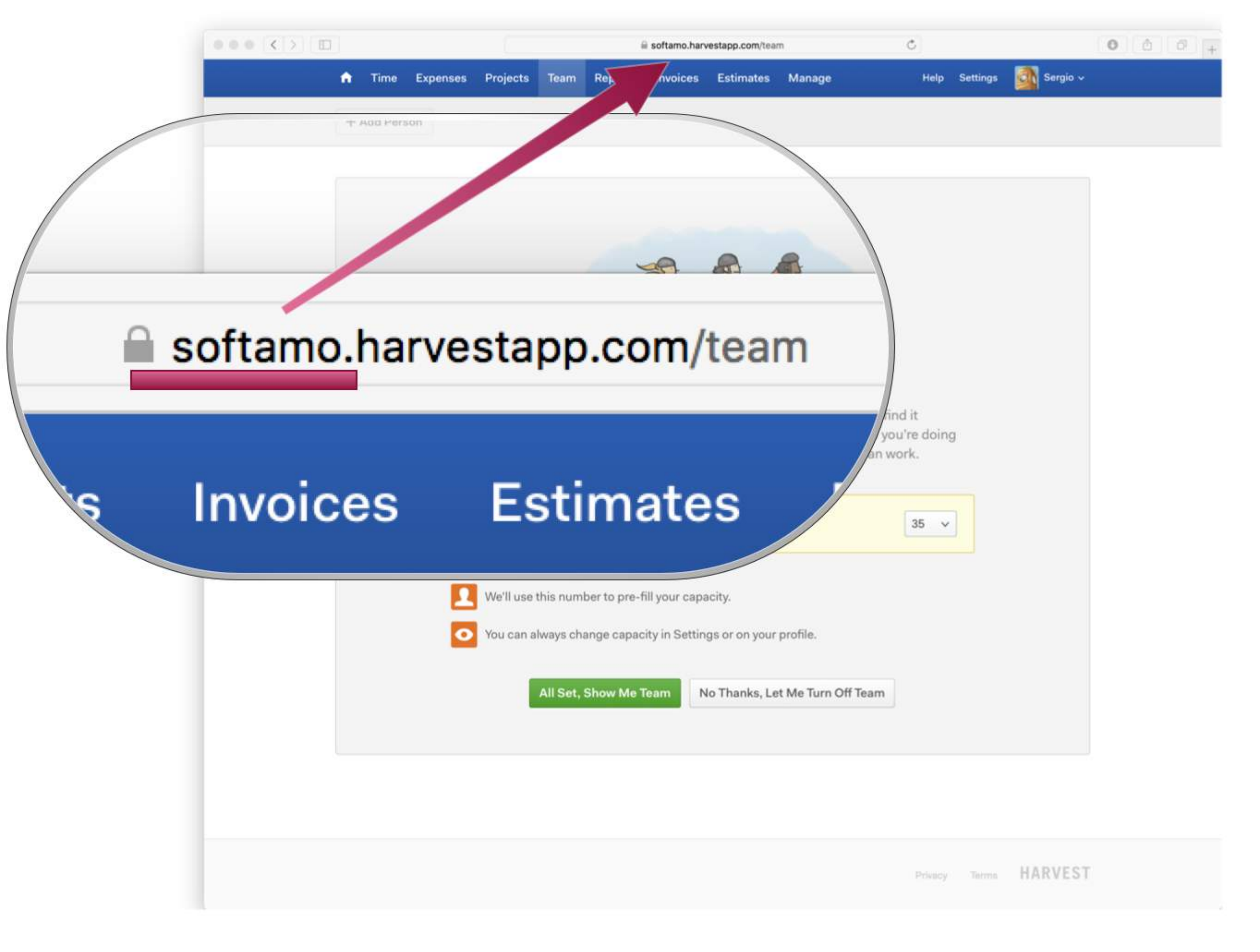
HOME TO GRAILS & MICRONAUT

TENANT RESOLUTION

BUILT-IN TENANT RESOLVERS

Name	Description
Cookie Tenant Resolver	Resolves the current tenant from an HTTP Cookie
HttpHeaderTenantResolver	Resolves the current tenant from the request HTTP Header
PrincipalTenantResolver	Resolves the current tenant from the authenticated username.
Subdomain Tenant Resolver	Resolves the tenant id from the sub domain
SessionTenantResolver	Resolves the current tenant from the HTTP Session.

SUBDOMAIN TENANT RESOLUTION

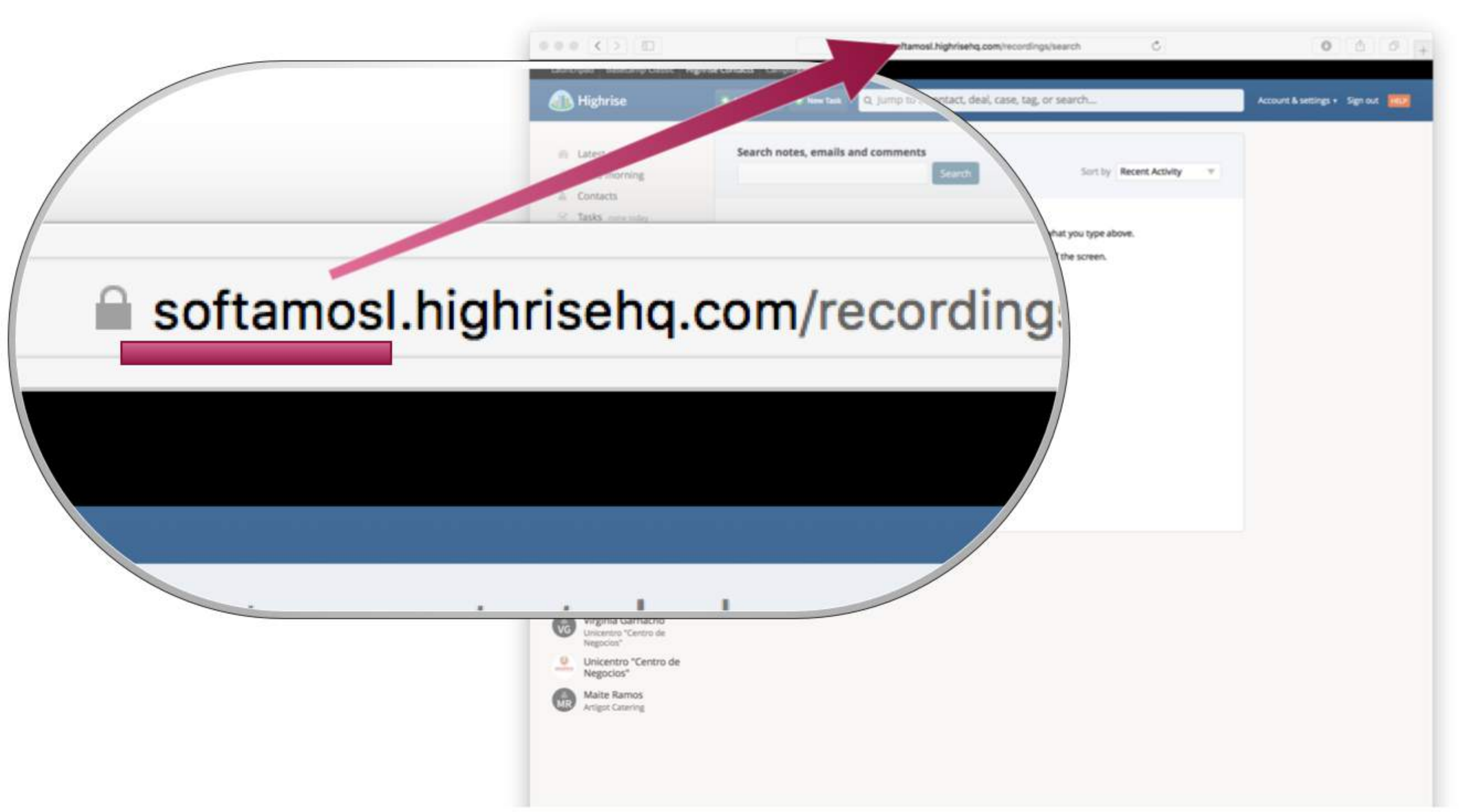


softamo.harvestapp.com/team

We'll use this number to pre-fill your capacity.

You can always change capacity in Settings or on your profile.

SUBDOMAIN TENANT RESOLUTION



softamosl.highrisehq.com/recording

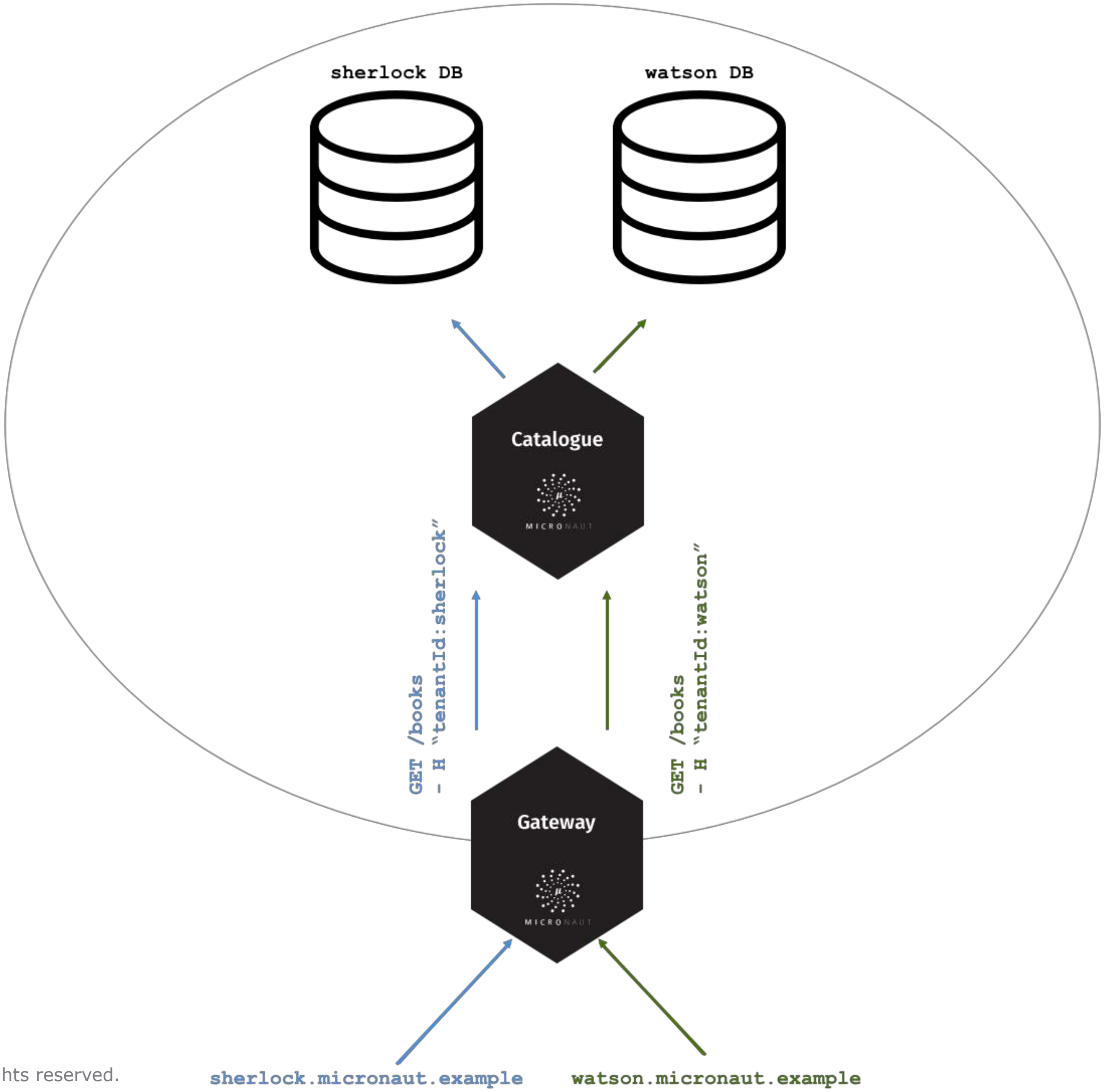
...ftamosl.highrisehq.com/recordings/search

BUILT-IN TENANT RESOLVER

Name	Description
SystemPropertyTenantResolver	Resolves the tenant id from a system property
FixedTenantResolver	Resolves against a fixed tenant id.

OBJECT COMPUTING
HOME TO GRAILS & MICRONAUT

TENANT PROPAGATION



sherlock DB

watson DB

Catalogue

MICRONAUT

GET /books
- H "tenantId:sherlock"

GET /books
- H "tenantId:watson"

Gateway

MICRONAUT

sherlock.micronaut.example

watson.micronaut.example

MULTI-TENANCY PROPAGATION

catalogue/src/main/resources/application.yml

```
micronaut:
  multitenancy:
    tenantresolver:
      httpheader:
        enabled: true
```

MULTI-TENANCY PROPAGATION

gateway/src/main/resources/application.yml

```
micronaut:
  multitenancy:
    propagation:
      enabled: true
      service-id-regex: 'catalogue'
    tenantresolver:
      subdomain:
        enabled: true
    tenantwriter:
      httpheader:
        enabled: true
```

DEMO

GORM MULTITENANCY

https://gorm.grails.org/latest/hibernate/manual/index.html#multiTenancy

GORM Multi-tenancy

- Configure Multitenancy Mode
- Configure Tenant Resolver
- Configure Domain classes which you want to be regarded as multi tenant.
- User `Tenants.*` or Multi-tenancy Transformations.

Multi-tenancy modes

Modes	Description
DATABASE	Separate database with a separate connection pool is used to store each tenants data.
SCHEMA	The same database, but different schemas are used to store each tenants' data.
DISCRIMINATOR	The same database is used with a discriminator used to partition and isolate data.

Multi-tenancy modes - Data isolation

Modes	Isolation
DATABASE	🔒🔒🔒
SCHEMA	🔒🔒
DISCRIMINATOR	🔒

MULTI-TENANCY PROPAGATION

src/main/resources/application.yml

```
grails:
  gorm:
    multiTenancy:
      mode: DATABASE
      tenantResolverClass: 'io.micronaut.multitenancy.gorm.HttpHeaderTenantResolver'
```

DOMAINS MULTI-TENANCY

```
@Entity
class Book implements MultiTenant<Book> {
    String title
}
```

DOMAINS - PARTITIONED MULTI-TENANCY

```
@Entity
class Book implements MultiTenant<Book> {
    String title
    String tenantId
}
```

DOMAINS - PARTITIONED MULTI-TENANCY

```
@Entity
class Book implements MultiTenant<Book> {
    String title
    String publisher
    static mapping = {
        tenantId name: 'publisher'
    }
}
```

Multi-tenancy transformations

Transformation	Description
@CurrentTenant	Resolve the current tenant for the context of a class or method
@Tenant	Use a specifc tenant for the context of a class or method
@WithoutTenant	Execute logic without a specific tentnat (using the default connection)

DOMAINS - PARTITIONED MULTI-TENANCY

```
import grails.gorm.multitenancy.*

// resolve the current tenant for every method
@CurrentTenant
@ReadOnly
class TeamService {

    // execute the countPlayers method without a tenant id
    @WithoutTenant
    int countPlayers() {
        Player.count()
    }

    // use the tenant id "another" for all GORM logic within the method
    @Tenant({"another"})
    List<Team> allTwoTeams() {
        Team.list()
    }

    List<Team> listTeams() {
        Team.list(max:10)
    }

    @Transactional
    void addTeam(String name) {
        new Team(name:name).save(flush:true)
    }
}
```

GORM DEMO

LEARN MORE ABOUT OCI EVENTS & TRAINING

Events:

- objectcomputing.com/events

Training:

- objectcomputing.com/training
- grailstraining.com
- micronauttraining.com

Or email info@ocitraining.com to schedule a custom training program for your team online, on site, or in our state-of-the-art, Midwest training lab.

OBJECT COMPUTING

HOME TO GRAILS & MICRONAUT

CONNECT WITH US

☎ 1+ (314) 579-0066

@objectcomputing

objectcomputing.com