

Sistemas de movilidad para aportar más valor: Tryvy

Sergio Morillo Navarro



¿Qué es Tryvy?

Plataforma para simplificar desarrollo de apps basadas en Tryton

- ***Tryton (www.tryton.org)***

Is a three-tier high-level general purpose application platform under the license GPL-3 written in Python and using PostgreSQL as database engine.

- ***Kivy (www.kivy.org)***

Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps

- ***KivyMD (gitlab.com/kivymd/KivyMD)***

Set of widgets for Kivy inspired by Google's Material Design

¿Por qué Tryvy?

- *No hay solución “estándar” de movilidad en Tryton*
- *Opciones habituales:*
 - App Web responsive: UX?
 - App nativa (Java): duplicidad de código
- *No python no party*
- *Funciona en android, iOS, Windows, Linux, MacOS*

¿Por qué Tryvy?

- *Utiliza toda la potencia del framework Tryton*
 - Gestión usuarios
 - Control acceso a datos
 - Permisos
 - Reglas de filtrado
 - Reutilización código Tryton en cliente
 - PYTHON, defaults, on_changes, ...
- *Funciona SIN conectividad*

¿Qué ofrece Tryvy?

- *Simplifica desarrollo*
- *Formularios predefinidos*
 - login, menú, ajustes
- *Formularios “tipo”*
 - lista, ficha, ficha con pestañas, diálogos, ...
- *Widgets “tipo”*
 - texto, fecha, hora, dato asociado (m2o), ...
- *Gestión de errores*

¿Qué ofrece Tryvy?

- *Filtros “tipo”*
 - por fecha, texto, ...
- *Botones “tipo”*
 - flotante, en barra navegación
- *Paginación en listas (swipe up)*
- *Refresco de datos en listas (swipe down)*
- *Multi-idioma*
- *2 modos de trabajo: Online vs Offline*

¿Qué ofrece Tryvy?

- *Código .py y .kv para crear un formulario lista*

```
3 from tryvy.ui.tree import TreeAvatarItem
4
5
6 class SaleTreeItem(TreeAvatarItem):
7     pass
```

```
<SaleTreeItem>
first_record_text: self.record['rec_name'].value
second_record_text: self.record['shipment_address.rec_name'].value
AvatarLabel:
...
```

¿Qué ofrece Tryvy?

- *Código .py y .kv para crear un formulario ficha*

```
from tryvy.ui.form import FormItem

class SaleItem(FormItem):
    _name = 'sale.sale'
```

```
<SaleItem>:
  TrytonCharField:
    field_icon: 'format-list-numbers'
    field_name: 'number'
  TrytonM20IconField:
    field_icon: 'city'
    field_name: 'company'
  TrytonDate:
    field_icon: 'calendar'
    field_name: 'sale_date'
  TrytonM20IconField:
    field_icon: 'account'
    field_name: 'party'
  TrytonM20IconField:
    field_icon: 'account-location'
    field_name: 'shipment_address'
  TrytonM20IconField:
    field_icon: 'account-location'
    field_name: 'invoice_address'
  TrytonM20IconField:
    field_icon: 'credit-card'
    field_name: 'payment_term'
  TrytonM20IconField:
    field_icon: 'currency-usd'
    field_name: 'currency'
```


¿Cómo funciona Tryvy offline?

- *Servidor tryton (sqlite) en dispositivo*
- *Sincronización servidor “replica” y servidor central*
 - indicamos modelos a sincronizar

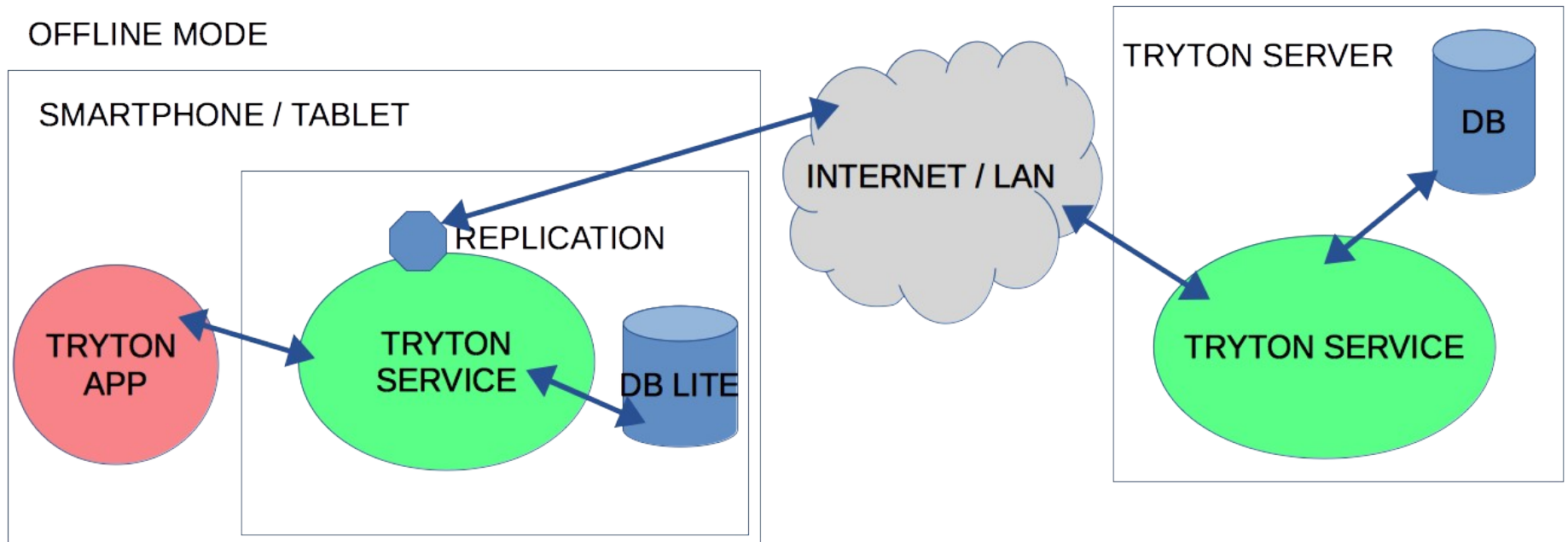
```
4 from tryton_replication import ReplLocalIdDestMixin
5
6
7 class Party(ReplLocalIdDestMixin, metaclass=PoolMeta):
8     __name__ = 'party.party'
9
```

replica side

```
4 from tryton_replication import ReplLocalIdSrcMixin
5
6
7 class Party(ReplLocalIdSrcMixin, metaclass=PoolMeta):
8     __name__ = 'party.party'
9
```

server side

¿Cómo funciona Tryvy offline?



Dispositivo trabaja con datos en local

Servicio que replica datos de manera incremental

¿Algún ejemplo?

The logo for Datalife, featuring a stylized white icon of a curved arrow pointing upwards and to the right, followed by the word "Datalife" in a white, italicized, sans-serif font.

¡Gracias! ¿Preguntas?

www.datalifeit.es

info@datalifeit.es

bitbucket.org/datalife_sco/projects/TRYVY

