EU Horizon 2020 Research & Innovation Program
Digital transformation in Health and Care
SC1-DTH-06-2020
Grant Agreement No. 101016496

SimCardioTest - Simulation of Cardiac Devices & Drugs for in-silico Testing and Certification

Technical Report
D 4.2 Software API for pharmacokinetics

Work Package 4 (WP 4)
Use Case 3 - Drug Efficacy & Cardiotoxicity

Task Lead: ExactCure, France
WP Lead: UPV, Spain

PUBLIC

DELIVERABLE INFORMATION

Deliverable number	D4.2
Deliverable title	Software API for pharmacokinetics
Description	This document describes API specification
Lead authors	Maika Warita (ExactCure)
Contributors	Léo Hubert, Samuel Baroudi, Kévin Koloskoff, Sylvain Benito (ExactCure)
Due date	30 June 2022
Submission date	30 June 2022
Comments	

Document history			
Date	**Version**	**Author(s)**	**Comments**
14 June 2022		M. Warita	Initial Draft
15 June 2022	V1	M. Warita	Finalization with contributors
15 November 2022		M. Barbier	Format edition

Table of Contents

EXECUTIVE SUMMARY

In SimCardioTest, ExactCure (EXC) share their expertise in personalized pharmacokinetics/ pharmacodynamics (PK/PD) modelling. ExactCure developed API to facilitate SimCardioTest project to integrate PK simulation in the pipeline of the cloud-based platform. The first version of API was delivered in October 2021 and was successfully integrated into the demo system implemented by InSilicoTrials in WP1. The list of molecules, to which simulation service will be provided on API, was carefully selected by studying cardiovascular effect through the discussion with WP4 team members. The system is designed to allow the project to add and update models simply by placing them in the database when they are ready. Results of the simulations from some models have started to be used in subsequent tasks in WP4.Technical concern about time conversion and scale was identified through executing this task. The impact of this issue should be discussed with the concrete use case.

1- Background

SimCardioTest will develop a standardised and secure cloud-based platform where in-silico trials run seamlessly. Three cardiac use cases will demonstrate the platform effectiveness, along with the required verification & validation processes and certification support of the medical device or medicine.

In this project, ExactCure shares its expertise in personalized PK/PD modelling. ExactCure is the task leader of T4.1 "Development of personalised pharmacokinetic models" of WP4 "Use Case 3 - Drug Efficacy & Cardiotoxicity".

1. Mine the literature to establish personalized models of the drugs targeted in the project.
2. Simulate the concentration of the drug in the body of the patient, taking into account his/her personal parameters (weight, gender, renal status, etc.)
3. Provide the resulting concentration as an input to other consortium partners, who will use it to run their own simulations at heart and cell levels, assuming that the drug concentration at the surface of the cells is the same as the free plasma concentration.

This personalized pharmacokinetics model simulation service is provided through API so that it can be used by the partners in the project to execute subsequent tasks, T4.2 "Simulation of drug effects on cardiac electromechanics", T4.3 "In-silico assessment of drug safety & cardiotoxicity" and T4.4 "In-silico assessment of drug efficacy". It is integrated into the cloud-based platform which is developed in WP1 as well. This technical report describes the specifications of this API.

2- Requirements analysis

This API fulfil these requirements from the project.

Category	Title	Requirement	Link
Simulation	Model Coverage	Should simulate a wide range of molecules with cardiovascular effect. The list of molecules was studied and agreed in UC3.	See section 5
	Personalization	The goal is to simulate the effect of age and gender if relevant.	See section 5
	Time Series Data	API should return time series of drug concentration data so that UPV can use the output for the cardiac model.	
Infrastructure	REST API	API should be implemented to facilitate the integration with the platform. We chose REST as it is one of the most common kinds of web services available today	
	Security (1)	Only project members can access to the simulation service.	
	Security (2)	Project members can access to the simulation of defined molecules, not to all the molecules in ExactCure portfolio.	
	Down Time	The service is supposed to be accessible 365 days 24 hours.	

This API is not fully optimized for the following points and open for adjustments, if necessary, when the project will finalize the orchestration of the platform.

Category	Title	Requirement	Link
Simulation	Time Scale Optimization	The cardiac models simulation is in milliseconds, whereas PAPI returns time series data in the scale in seconds. Special attention should be paid when we connect two simulations on the platform.	
Usability	Model Search by Molecule	Currently, simulation endpoint is provided for drugs in France. For international users, we need an endpoint with which we can call the simulation with molecule name, route, galenic form and dose.	
	Language	In the future, when we can search a model by the molecule name, the function to query the molecule in local language would be nice to have for usability.	
Infrastructure	Traffic Size	Technical requirements for this API would be reconsidered when the platform will go live.	

3- Functional specification

Name

Partner's API (PAPI)

Overview

PAPI is a RESTful API that allows partners to access to the ExactCure's Services.
The endpoints allow users to read the contents but not allow them to create, update and delete it.

Host	Protocol	Version	Data Format
https://partners-plive.exactcure.com/api	https	V2	JSON

General documentation is available online.
https://partners-dev.exactcure.com/docs (How to use)
https://partners-plive.exactcure.com/api/v2/docs# (Demo)

Access Token

Access token was provided to the users in the project.
The documentation pages can be accessed by anyone. The use of the API is controlled by a bearer token. Users with the token can query entire drug database. Portfolio of drugs and molecules which users can simulate is controlled by the token.

Ex: Users from SimCardioTest can see that 'Paracetamol' is in our database but cannot simulate it as Paracetamol is not in the portfolio.

Please refer to section 5for the list of molecules allowed for SimCardioTest project users.

Endpoints

This API is developed to facilitate the integration of ExactCure's service into the partners' application. The API provides various services as follows.

- Search engine for medicines
- Look up necessary covariates to run the simulation
- Simulation for given drugs, posology, and patient's profile (covariate)
- Simulations
- Qualitative Drug-Drug Interaction Information
- Information of drugs (risk, storage time etc.)

Though SimCardioTest PJ can access to the all the services with the provided token, please do not use the endpoints other than indicated in the following list. The endpoints relevant for SimCardioTest Project are follows.

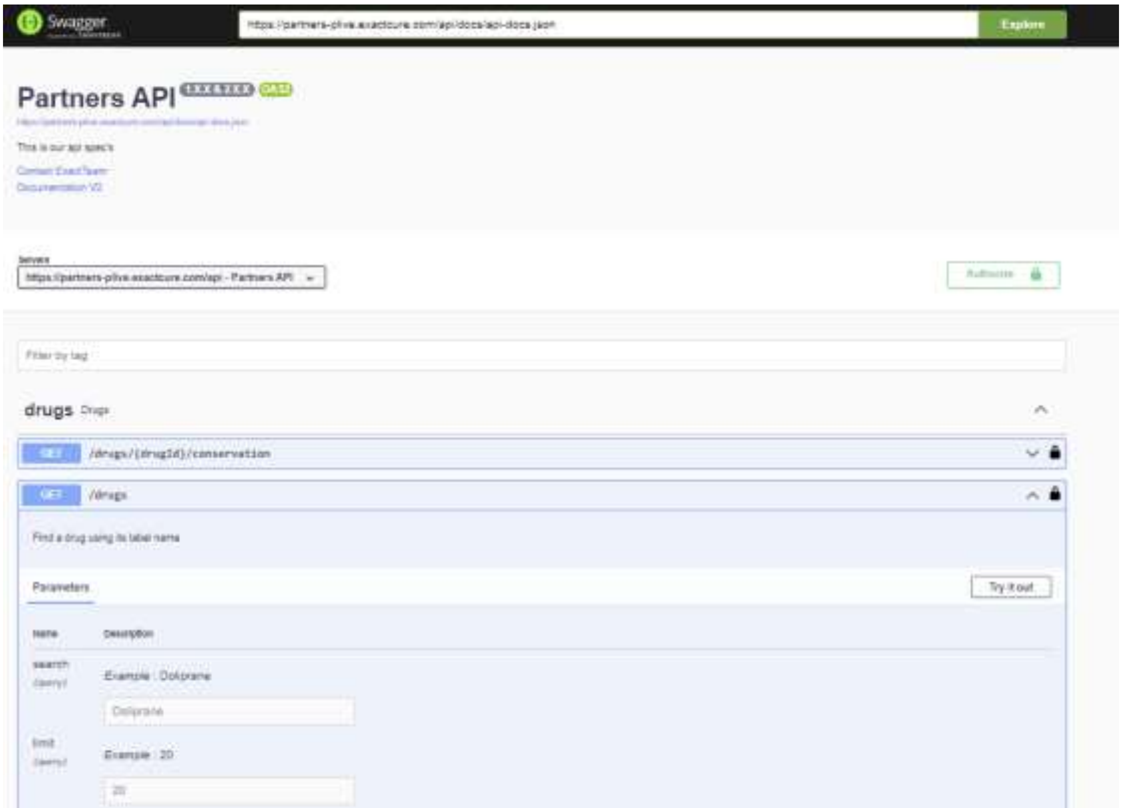
Method	Endpoint	What is it for?
GET	/drugs	To query drug_id (identifier of drugs)
GET	/drugs/covariates/{drugId}	To look up the necessary covariate to run the simulation of the drug.
GET	/drugs/covariates	To get a list of all the definition of covariates. For example, Child Pugh Class is defined as a categorical variable with modality 'healthy', 'A', 'B' and 'C'.
POST	/v2/simulations	By posting subject's (patient's) covariate, prescription data (drug_id), intake start time, quantity and interval, this endpoint kick our simulation engine and returns time concentration data.

Status Code

Code	Details	
200	Successful operation	
401	Error: Unauthorized	Token is not valid.
422	Bad request, wrong request body parameters	The body parameter is not written correctly.

User Interface

The API equips user interface generated by Swagger UI. Non-developers can use API service if the user has data literacy, such as reading and writing json. A tutorial documentation was provided from EXC members to WP4 members.



Partners API

This is our api spec's

Contact ExactTeam

Documentation V2

Find a drug using its label name

Example : Doliprane

Example : 20

Time Conversion

The output from endpoint /simulation contains an array of timestamps of Unix Epoch time. For example, time 1654812000 means Thursday, June 9, 2022 10:00:00 PM in GMT. This conversion is undertaken by PAPI.

```
{
 "data": [
  {
   "molecule": {
    "id": 555,
    "name": "nifedipine",
    "label": "Nifedipine"
   },
   "times": [
    1654812000,
    1654812000,
    1654812000,
    1654812000,
    1654812003,
    1654812039,
    1654812399,
    1654815999,
    1654840800,
    1654840800,
```

As the reader would see in the example, some elements of ‘times’ array have equal value. This is the technical limitation of using Unix Epoch time, which minimum time scale is in seconds. Our calculation engine, described in section 6, returns time series data in hours and the gap between two points can be less than a second. This conversion is also undertaken by PAPI.

This case can be observed anywhere in the simulation, but usually near the administration time of the drug. Fast kinetics, like infusion or IV administrations, tend to provoke this phenomenon, since the computing core will detail those parts of the simulation output much more, resulting in points separated only by a few milliseconds. This is the reason why several elements of times array are converted into the timestamp with the same value.

This is not an issue when the simulation result is used and interpreted by medical professionals at the hospitals because their interest is not in studying PK in time scale of seconds. But for SimCardioTest project, since this PK simulation will be an input of successive cardiology model which is simulated in milliseconds scale. This point should be carefully discussed with the concrete use case.

4- How to use

Drug Identifier

ExactCure maintains French drug database. The drugs are mapped to component molecules and relevant pharmacokinetics models. We define in-house drug_id which is sequential numbers starting from 1.

*In the future, the service will cover drugs data in other countries as well as an endpoint to execute simulation with a molecule name and other necessary data which will impact pharmacokinetics, such as administration route and galenic forms. But as of today, user need to specify French drug to simulate a molecule of user's interest.

To search drug_id, the user can use endpoint /drugs.
The user can query drug_id by passing query or cip code as a parameter.
For example, to search nifedipine drugs, user can request data by /drugs?query=nifedipine

Object 'id' is drug_id.

```
{
 "data": [
  {
    "id": 8828,
    "uid": "66932763",
    "label": "NIFEDIPINE ARROW L.P. 30 mg, comprimé pelliculé à libération prolongée",
    "url": "https://base-donnees-
publique.medicaments.gouv.fr/affichageDoc.php?typedoc=N&specid=66932763",
    "unit": "mg",
    "quantity": 30,
    "format": "un comprimé",
    "molecules": [
     {
        "name": "nifedipine",
        "label": "Nifedipine",
        "quantity": 30
     }
    ],
    "language": "fr",
    "country": "FR",
    "pathology": "HYPERTENSION",
    "therapeutic_class": "ANTI_HYPERTENSION",
    "is_simulable": true,
    "is_topical": false,
    "is_otc": false,
    "is_chronic": false,
    "is_authorized": true
  }
```

Body Parameter

Here we describe about simulation endpoint (/api/v2/simulations) body parameter since the user is required to fill ‘subject’ and ‘prescription’ key with defined rule.

This is an example of the complete request's structure.

```
{
  "params": {
    "locale": "fr"
  },
  "subject": {
    "birthday": "1976-09-20",
    "sex": "M",
    "weight": 70.5,
    "height": 175.5
  },
  "prescription": {
    "drug": {
      "id_type": "id",
      "id": "123"
    },
    "start_date": "1997-07-10T00:00:00+02:00",
    "end_date": "1997-07-17T00:00:00+02:00",
    "intakes": [
      {
        "quantity": 1.5,
        "time": "08:12",
        "interval": 6
      }
    ]
  }
}
```

Subject

This object contains the patient's information like birthday, weight, height, and renal statuses. In our system, this patient profile is called **'covariate'**. Necessary covariates depend on drugs (For drug A, you need 'weight' and 'age' but for drug B you need 'weight' and 'child pugh class'). And the value of the covariate should be filled in the body in the given rule. For example, weight should be provided in kilograms not in pound.

What are necessary covariates for a simulation of the drug?

To know what should be in this object, the user needs to look up covariate by using `/drugs/covariates/{drugId}`

Here is the example of Nifedipine. `cov_key` must be used to describe subject profile.

```
{
  "data": {
    "covariates": [
      {
        "description": {
          "name": "Âge",
          "unit": "ans",
          "placeholder": "Saisissez votre âge"
        },
        "position": null,
        "comment": "formula is an exception managed directly by ExaTwin",
        "covariate_group_id": 1,
        "cov_key": "age",
        "cov_type": "experimental",
        "cov_spec": {
          "type": "float",
          "unit": "years",
          "default_value": 40,
          "min": 0,
          "decimal_digits": 2
        },
        "cov_rule": null
      },
      {
        "description": {
          "name": "Classe de Child-Pugh",
          "placeholder": "Sélectionnez votre classe de Child-Pugh",
          "help_name": "Qu'est-ce que ma classe de Child-Pugh ?",
          "help_text": "Pour en savoir plus sur votre classe de Child-Pugh, demandez conseil à votre
médecin.",
          "modality_translation": {
            "healthy": "Pas d'insuffisance hépatique",
            "A": "Classe A",
            "B": "Classe B",
            "C": "Classe C"
          }
        },
        "position": 2,
        "comment": null,
      "covariate_group_id": 4,
      "cov_key": "child_pugh_class",
      "cov_type": "additional",
      "cov_spec": {
        "type": "categorical",
        "unit": null,
        "modality": [
          "healthy",
          "A",
          "B",
          "C"
        ],
        "has_priority_on": [
          "hepatic_status"
        ]
      },
      "cov_rule": [
        {
          "input": [
            "child_pugh_score"
          ],
          "equation": "Piecewise((healthy, child_pugh_score < 5), (A, child_pugh_score < 7), (B,
child_pugh_score < 10), (C, child_pugh_score < 16))"
        },
        {
          "input": [
            "hepatic_status"
          ],
          "equation": "Piecewise((healthy, normal), (A, mild), (B, moderate), (C, severe))"
        }
      ]
    },
    {
      "description": {
        "name": "Poids",
        "unit": "kg",
        "placeholder": "Saisissez votre poids"
      },
      "position": 4,
      "comment": null,
      "covariate_group_id": 1,
      "cov_key": "weight",
      "cov_type": "mandatory",
      "cov_spec": {
        "type": "float",
        "unit": "kg",
        "default_value": 70,
        "min": 0.1,
        "max": 500,
        "decimal_digits": 2
      },
      "cov_rule": null
    }
  ],
  "covariate_groups": [
    {
        "id": 4,
        "description": {
          "name": "Informations hépatiques"
        },
        "position": 4
      },
      {
        "id": 1,
        "description": {
          "name": "Informations générales"
        },
        "position": 1
      }
    ]
  }
}
```

How to know covariate description rule?

The description object contains the name, unit and placeholder of the covariate. This object can be used to generate a form in the platform to guide the users to fill covariate information with expected manner. For some of the covariates you can find help_name and help_text keys in the description object. The position can help you to place the covariate in the form. The covariate_group_id is the id of the covariate group.

The value of cov_key is the text you need to provide as a key in the subject object in simulation request body. The cov_spec object contains the specification of the covariate:

- type: The type of covariate. (int, float, string, categorical, etc...)
- unit: The unit of the covariate. (For example: kg)
- default_value: The default value of the covariate. (For example: 70)
- min: The minimum allowed value of the covariate. (For example: 0.1)
- max: The maximum allowed value of the covariate. (For example: 500)
- decimal_digits: The number of decimal digits of the covariate. (For example: 2)

Depending on the type of the covariate, you can have more keys in the cov_spec object. For example, if the covariate is a categorical, you will find following keys:

modality: The list of values of the covariate. (for example: ["M", "F"])

default_value: The default value of the covariate. (for example: "M") You can find the translation of the modality values in description object as modality_translation key. (For example: ["M": "Homme"])

Prescription

In this object, prescription information should be provided. To perform a simulation, drug identifier, start date and end date of the treatment and intake time and quantity are required.

Key ‘drug’

Please key in the identifier of the drug the user wants to simulate. There are 2 ways to specify this, (exactcure) drug_id and cis code from French government.

CASE drug_id
If you work with drug_id, please write request body as follows.
Example:

```
{
   ...,
   "prescription": {
      "drug": {
         "id_type": "id",
         "id": "123"
      },
      ...
   }
}
```

If you work with cis code (French drug identifier from the government), please write request body as follows.
Example:

```
{
   ...,
   "prescription": {
      "drug": {
         "id_type": "cis",
         "id": "123456"
      },
      ...
   }
}
```

Key ‘start_date’ and ‘end_date’

The start_date and end_date keys are required and must be in the format YYYY-MM-DDTHH:mm:ss+HH:mm. (ISO 8601) These dates will define the simulation's period (window), and will return simulations points between these two dates.

Key ‘Intakes’

The service assumes that a patient would intake drugs several times, for example ‘1 tablet, 3 times a day for 5 days’.Each ‘intake’ should contain information about administration time and quantity of the drug.

Required 'Intakes' information depend on the drug's intake type.

As of 14 June 2022, we have two types of intakes:

Default (Bolus, oral, rectal etc.)
Infusion (infusion)

Default intake format

For each intake, the users are expected to fill the following keys:

- quantity: The quantity of the drug that the patient will take. (In tablet) or
- dose: The dose of the drug that the patient will take. (In mg)
- time: The time of the drug that the patient will take. (Hamm time) or
- date: The date of the drug that the patient will take. (in YYYY-MM-DDTHH:mm:ss+HH:mm format)

interval: The interval between intakes. (In hour)

The interval field is optional and by default set to 24 hours.

Example with dose and time: (Example on PARACETAMOL 1000mg)

```
{
   ...,
   "prescription": {
      ...,
      "intakes": [
         {
            "dose": 1000,
            "time": "08:12",
            "interval": 6 // optional
         }
      ]
   }
}
```

Example with quantity and date without interval: (Example on PARACETAMOL 1000mg)

```
{
   ...,
   "prescription": {
      ...,
      "intakes": [
         {
```

```
        "quantity": 1,
        "date": "1997-07-10T00:00:00+02:00"
      }
    ]
  }
}
```

Infusion intake format

Allowed intakes keys for infusion are:

- dose (in mg)
- time or date
- duration: The duration of the infusion. (In hour)
- rate: The rate of the infusion. (In mg/hour)
- interval (optional and by default to 0, meaning that the infusion will not be repeated)

Note that a minimum of two of these keys must be present:

- dose (required)
- duration
- rate

If you provide all the three keys, please be sure to provide consistent values. Else, the API will return an error with advice on how to fix the problem.

How to plot output

In /simulation output, you will get an array of times with the exact same length as the simulation points, which can be found in the results -> curves -> values array, as shown below:

```
{
  "data": [
    {
      "times": [
        ...,
      ],
      "results": [
        {
          "curves": [
            {
              "values": [
                ...,
                33.78900270833991,
                94.60399728998861,
                215.97950117080086,
                262.66059480910457,
                281.31898088588684,
                288.78008867662885,
                ...,
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

You need to provide those two series (times and values) to a graph library to plot the simulation, with times in abscissa and values in ordinate. Some additional information is available in the results object, which is supposed to be used for graph (such as title, units etc...).

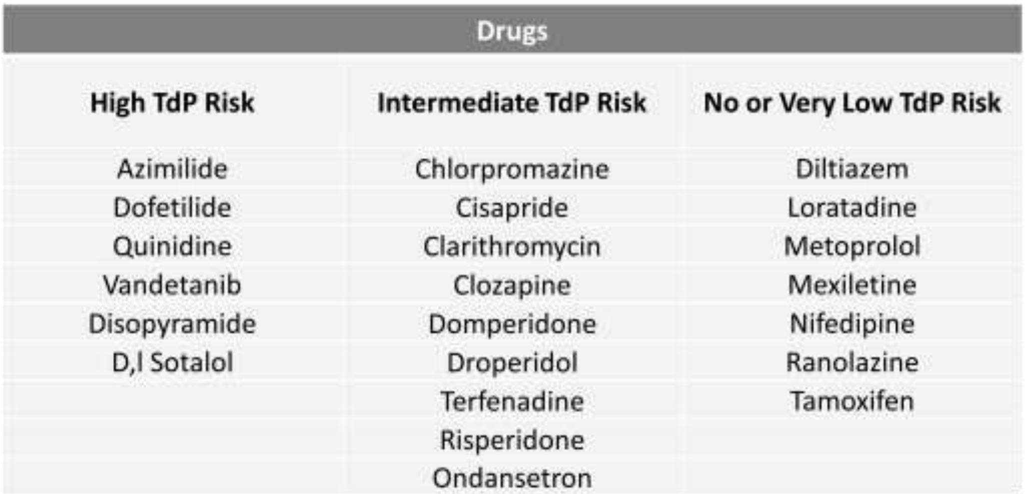
5- List of molecules

Molecules of Interest

Drugs available for SimCardioTest users have been decided between UPV and EXC, and these molecules correspond to the list on which the two partners have worked on so far.

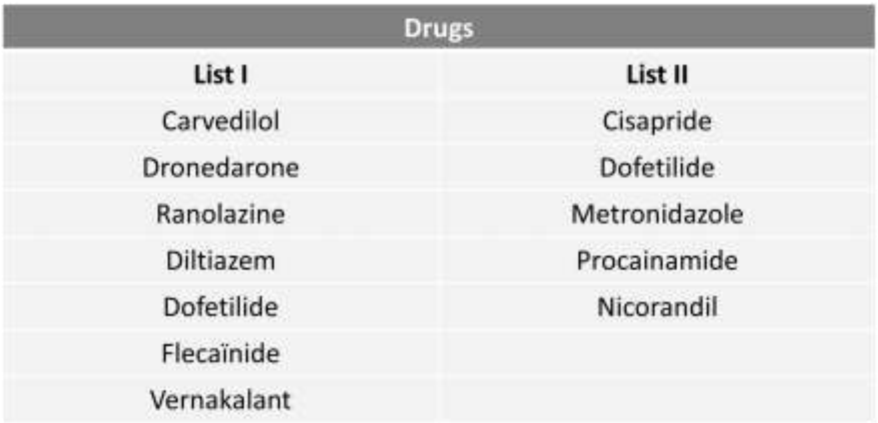
Scientifics first agreed to work on CiPA (The comprehensive in Vitro Proarrhythmia Assay) drugs to evaluate risks of torsade de pointes (TdP) induced by drugs with on / off target cardiac effect (see table 1).
It was also decided to work on efficacy with a list of drugs for atrial fibrillation and heart failure (see table 2).

Table 1: CiPA drugs list modeled.



Drugs		
High TdP Risk	Intermediate TdP Risk	No or Very Low TdP Risk
Azimilide	Chlorpromazine	Diltiazem
Dofetilide	Cisapride	Loratadine
Quinidine	Clarithromycin	Metoprolol
Vandetanib	Clozapine	Mexiletine
Disopyramide	Domperidone	Nifedipine
D,l Sotalol	Droperidol	Ranolazine
	Terfenadine	Tamoxifen
	Risperidone	
	Ondansetron	

Table 2: List of Efficacy drugs modeled.



Drugs	
List I	**List II**
Carvedilol	Cisapride
Dronedarone	Dofetilide
Ranolazine	Metronidazole
Diltiazem	Procainamide
Dofetilide	Nicorandil
Flecaïnide	
Vernakalant	

Molecules on API

Below is the list of molecules which the project members have access through API as of 14 June 2022.

These models have been validated through the process described below and tested on our application. When we find additional medical or scientific information which makes us to improve the models or makes us to doubt the validity of the models, we may update or withdraw the models.

Not all the molecules of interests are available on API as of 14 June 2022 since not all the models are validated through our validation process. Also, we did not need to put all the validated molecules to the database to create demo system. But the system is designed to allow the project to add and update models simply by putting models in the database. The left molecules of interest will be added to the database before the partners start simulating the models from the project pipeline.

Molecules	Covariates
atenolol	weight, glomerular_filtration_rate
chlorhydrate de loperamide	weight
chlorpromazine	weight,glomerular_filtration_rate
clarithromycine	weight,glomerular_filtration_rate
clozapine	sex
diltiazem	weight,age
disopyramide	weight,glomerular_filtration_rate
domperidone	glomerular_filtration_rate
droperidol	weight
escitalopram	weight,age,height,mutation_cyp2c19
loratadine	child_pugh_class
metoprolol	weight,sex,mutation_cyp2d6
nifedipine	weight,age,child_pugh_class
ondansetron	weight,age,sex
propranolol	weight,age
sotalol	weight,glomerular_filtration_rate
vandetanib	weight,renal_status

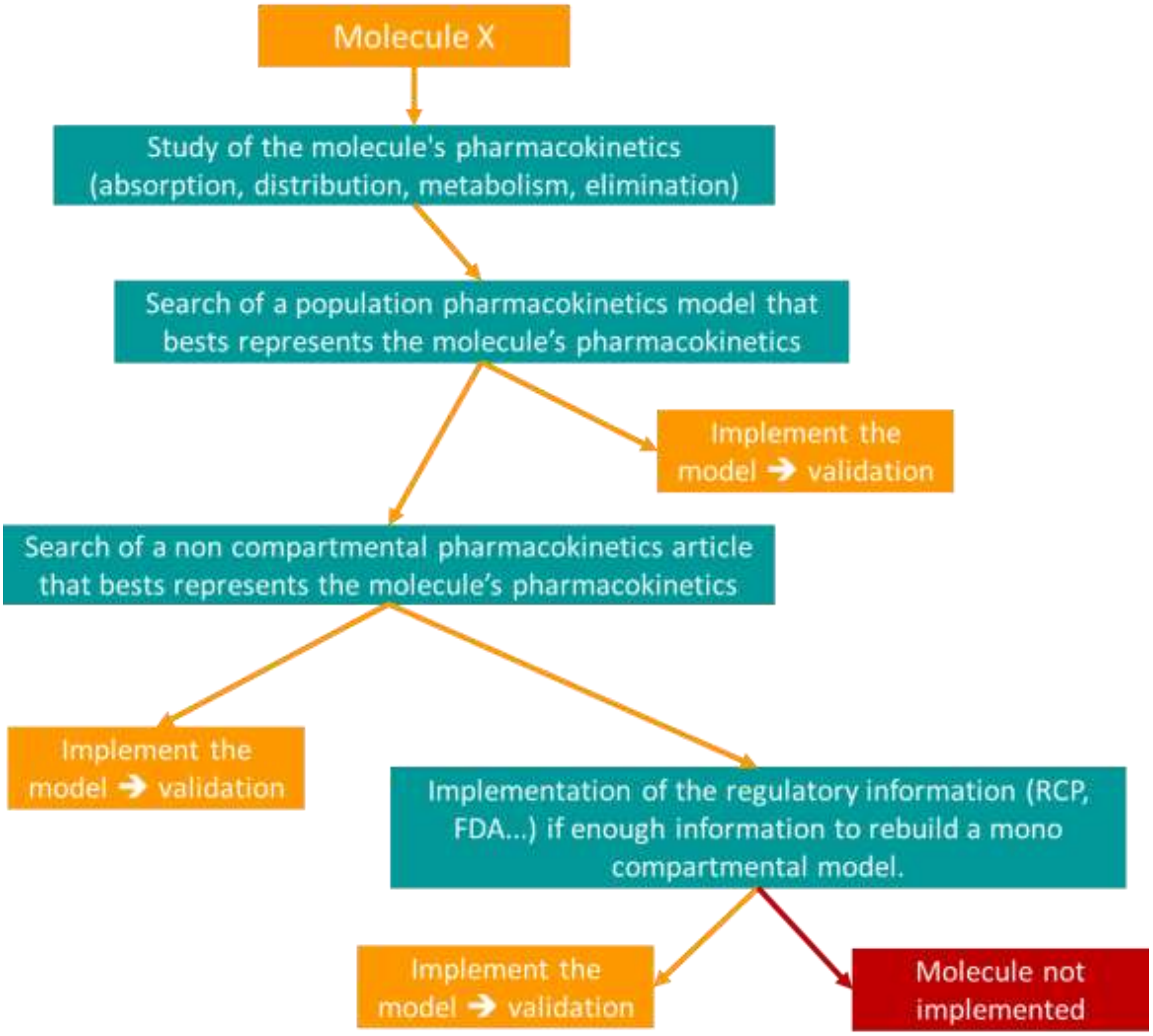
General Modelling Process

The sources available on a molecule's pharmacokinetics properties are as follows:

- Reglementary data (RCP on https://base-donnees-publique.medicaments.gouv.fr/, FDA data …).
- Compartmental models (population pharmacokinetics models on https://pubmed.ncbi.nlm.nih.gov/, https://www.researchgate.net/ or other sources).
- Noncompartmental pharmacokinetics data (https://pubmed.ncbi.nlm.nih.gov/, https://www.researchgate.net/ or other sources).

The guidance is to provide a model as close as possible of the molecule's pharmacokinetics. Example: If a molecule is indicated to have a biphasic elimination, then this should be modelled. However, if no data are available an approximation can be performed, through simpler modelling. This approximation will, anyway, must go through the validation process and be confronted to real exposure data.

The decision tree is as follows:

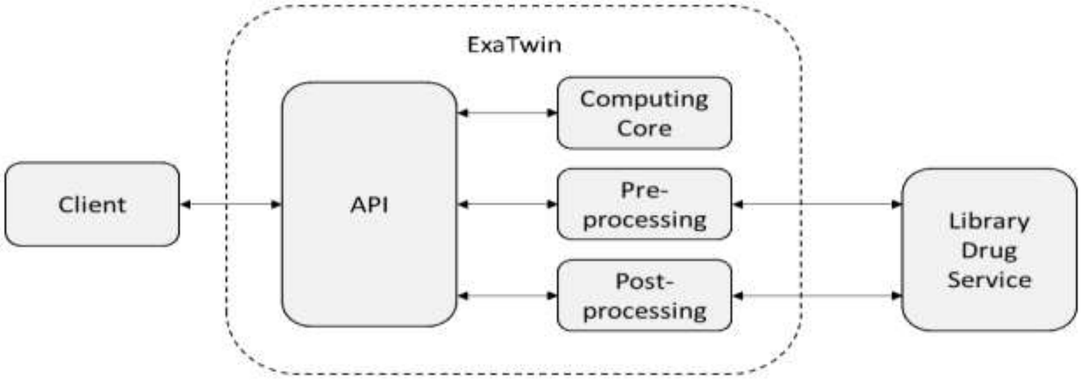

Molecule X
Study of the molecule's pharmacokinetics (absorption, distribution, metabolism, elimination)
Search of a population pharmacokinetics model that bests represents the molecule's pharmacokinetics
Implement the model → validation
Search of a non compartmental pharmacokinetics article that bests represents the molecule's pharmacokinetics
Implement the model → validation
Implementation of the regulatory information (RCP, FDA...) if enough information to rebuild a mono compartmental model.
Implement the model → validation
Molecule not implemented

Special Modelling Process for SimCardioTest Project

For SimCardioTest, the same process was followed except that it was focused on specific covariates that were of interest for the ongoing research. EXC and UPV agreed to work on the impact of gender, and renal status or age on the TdP risk and drug's efficacy. EXC tried to build models including these covariates when it has a significant impact on the pharmacokinetics of the drugs.

6- Calculation engine

When the user requests a simulation from PAPI, PAPI calls ExaTwin Simulation Service to perform it. ExaTwin is ExactCure's internal calculation service dedicated to the processing and simulation of pharmacometric models. Below is a description of the architecture of the service.In the diagram, PAPI is the Client.



Pharmacokinetics model data, before personalization, is accessed by ExaTwin from Library Drug Service. With subject (patient) covariate data, ExaTwin personalizes the model. Then, with the provided intake data, it predicts for example the time course of drug concentration in the patient and return it to PAPI as time series data. The conversion of covariate values (for example, birthday to age) is also managed by ExaTwin.

ExaTwin's detailed logic is not disclosed here as it is not the discussion point of WP4 (Development of personalized pharmacokinetic models) and it is involving future intellectual property of the company. The verification of ExaTwin is planned in WP6 (Verification, Validation, UQ & Certification) and details would be described and discussed there.

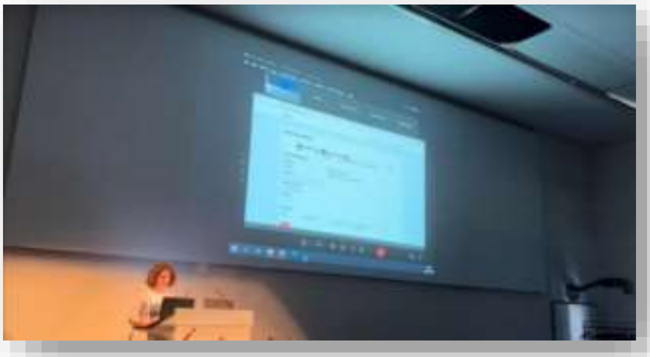
7- Results

The first version of PAPI was delivered in October 2021 and the connection to other subsequent tasks has been completed.

Demo

PAPI was successfully integrated into the platform demo and was presented at the General Assembly in March 2022 (picture below).

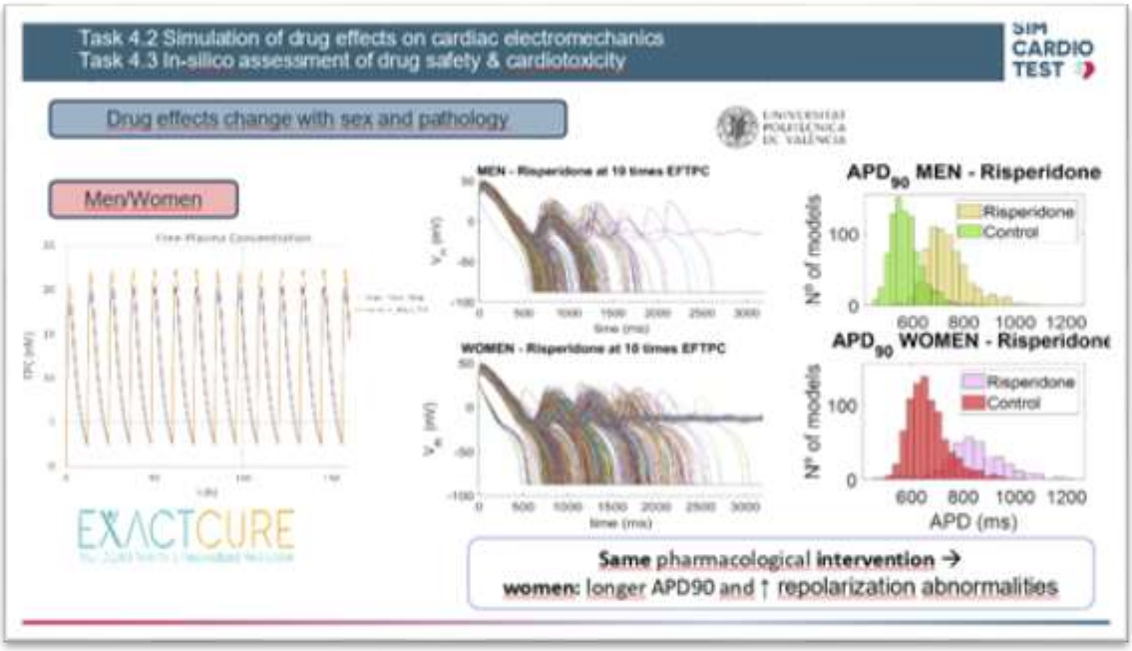
Connection to subsequent tasks in WP4



The API allows UPV to test different scenarios for the electrophysiological (EP) simulations. For tasks 4.3 (In-Silico assessment of drug safety & cardiotoxicity) and 4.4 (In-Silico assessment of drug Efficacy), UPV would be able to use the API to define a patient, select a drug, and obtain concentration/time series. But the learning cost of using API is not zero and it is essentially designed to be called from the platform. Thus, the simulations are performed by EXC team members in practice.

In that way, UPV is now able to use simulations service to obtain concentrations time series of drugs we agreed to work on. (See 5. List of molecules). The modelling team of EXC defines the scenarios (Patients/Drugs/dosing schedule) with UPV, and UPV can assess drug's efficacy, safety, and cardiotoxicity with the EP model.

Research is focusing on new insight such as the characterization of TdP risk for different populations (men vs women, healthy patients vs heart failure patients, etc.)



Task 4.2 Simulation of drug effects on cardiac electromechanics
Task 4.3 In-silico assessment of drug safety & cardiotoxicity

Drug effects change with sex and pathology

Men/Women

Same pharmacological **intervention** →
women: longer APD90 and ↑ repolarization abnormalities

*One of the slides presented in general assembly in March 2022 to present the project status by work package leader.

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016496