



# Sales Kit

## Taipy Internal Use and Partners

This document provides the various elements required to sell and present Taipy to a new prospect.

**Saleskit V 0.1 by CS Team**

# Table of contents

<b>Table of contents</b>	<b>2</b>
<b>Personae</b>	<b>3</b>
Selling to the CIO / CDAO	4
Recommendations	5
Selling to the Data Scientist	6
Questions to ask	6
Q1) What kind of tools/libraries are you currently using for the front end?	6
Q2) What tools/libraries are you using for the back end? Are you involved with data pipelines?	6
Q3) Are you using any Data platform?	7
Selling to the Python Developer	8
Questions to ask - On top of the ones for the Data Scientist	8
Selling to the BI Analyst / Head of BI unit	9
<b>Competition with Streamlit</b>	<b>10</b>
Taipy is more performant	10
On refresh	10
On the support of large datasets	10
Taipy supports async calls	10
Taipy provides MultiUser with broadcasting	10
Taipy integrates natively with Plotly (not streamlit)	11
Taipy provides full backend capabilities	11
Taipy has Taipy Designer	11
Taipy has a rich set of APIs	11
Taipy is customisable	12
Where does Streamlit win?	13
Streamlit has a larger community.	13
Streamlit is more connected to other graphical widget libraries.	13
Streamlit is free	13
Streamlit is backed by Snowflake	13
Streamlit is not really running on notebooks	13
<b>Competition with PowerBi/PowerApps and other BI tools (Tableau)</b>	<b>15</b>
<b>Demos</b>	<b>16</b>

# Personae

At this stage, we have identified four main personae through our various meetings.

Personae	Description
<i>CIO / CDAO</i>	Head of IT Dpt or Analytics Dpt. These Dpts may be at different proficiency levels of Python development. May also cover BI Activities.
<i>Data Scientist</i>	Develops ML/AI models using Python. Wants to easily/quickly display visualization related to its data (input or output of ML). Often exposed to Streamlit or Gradio.
<i>Python Developer / Usually part of the IT Department</i>	Involved with full projects. Part of the IT group in charge of productizing applications covering GUI/Backend and even dev ops.
<i>BI Analyst / Head of BI unit</i>	Involved mostly with BI tools like PowerBI, Tableau, Qlik. Can be part of a BI Dpt or embedded inside a Business Unit.
<i>Citizen Data Scientist</i>	Citizen Data Scientists typically use automated, or semi-automated data science and analytics tools without requiring extensive coding knowledge.

## Selling to the CIO / CDAO

Before starting a presentation, it is always good to ask the background of the company and the IT Group. Need to find out if:

- They do have Python developers
- They use a Data Platform such as Databricks, Dataiku, GCP/Colab, AWS/SageMaker or else.
- The types of development/applications they do: Dashboard, Data Analytics, BI, or AI, etc.

Always ask about their pain points.

You need to establish the benefits of using Taipy.

You can go through the slide presentation: main\_presentation.pptx

In the first slides, it is important to highlight the “raison d’être” of Taipy.

Taipy exists for a single purpose:

- Enable the IT / Data Dpts to easily develop and deploy Data applications all in Python.
- Taipy’s objective is to become THE standard platform for the development & deployment of any Python application.

The major elements to highlight.

- Taipy is a complete framework (GUI front end, back-end, Deployment) that caters to any type of application development.
- Taipy is easy to learn & use
- Taipy can scale in terms of performance
- Taipy provides product support
- Taipy provides security
- Taipy caters for different End-User Profiles
- Etc.

Taipy has been designed:

- To provide a single platform across the various silos (Data Engineering, Data Scientists, MLOps, IT Dev, DevOps, End-User)
- To use the same platform for Pilot and full-scale development/Deployment.
- To consider the end-year: Scenario Management, Great Interactive Graphics, ...

Use the two Intermarché slides (from the main presentation), which provide an example of huge savings in terms of project productivity, development costs, and the ROI of the Intermarché application.

Often, the CIO can be technical and have Python coding experience. Ask him about it, if she/he does know Python, then you may cover all/some or all of the [Selling to the Data Scientist](#).

Even if he doesn't have any Python programming background, it is good to show the standard demo examples (sales kit repository) quickly to demonstrate how easy it is.

Always keep enough time to demonstrate Taipy Designer, the feedback has always been very positive.

Often, the CIO also covers a BI Dpt. Then refer to [Selling to the BI Analyst / Head of BI unit](#).

At the end of the presentation, it is important to explain the scope of Taipy Enterprise. There is a slide highlighting the differences between the Open Source Version and Taipy Enterprise. The salient points (for Taipy Enterprise) are Product Support, Authentication, Authorisation (User Roles), Taipy Designer, and better performance (distributed computing, faster data management).

## Recommendations

If the CIO is interested, he may actually mention that he will see with his team and try the open-source version. No deadline is provided, you may somehow lose control on the next step.

One option is to highlight that Taipy Designer is only part of the Enterprise Version (usually Taipy Designer often impresses most CIOs) and that, as a next step, we could do:

- a half-day training of their technical team
- Organize an evaluation license (for Taipy Designer) limited in time.

Another option is to suggest that we build a pilot/demonstrator with their team to demonstrate our capabilities.

# Selling to the Data Scientist

## Questions to ask

A Data Scientist typically has a set of tools/libraries he is familiar with (usually in the Python stack).

Q1) What kind of tools/libraries are you currently using for the front end?

### Answers:

Very often, Streamlit, sometimes Gradio (and, very rarely, other libraries like Reflex or Shiny Python) will be mentioned.

Provide all the elements/demos throughout the presentation that show the superiority of Taipy over Streamlit (see [Competition with Streamlit](#)).

You can:

- Show him Taipy Designer and that with Taipy, Demos can also be built extremely quickly
- Unlike Gradio, Taipy can move from a simple demo to a more complex full-blown project.

If they mention Gradio:

Gradio targets AI/ML GUI only, which explains its strong connection with [huggingface](#). It is also easy to use.

Gradio emphasizes rapid prototyping and shareability of ML models with various inputs and outputs.

Reflex seems more scalable than Streamlit or Gradio, but we almost never encounter anyone using it...

Finally, Plotly Dash could be mentioned. Plotly Dash is not strictly a direct competitor since you need to do serious GUI coding to develop Plotly Dash applications. Taipy is designed for non-GUI specialists, in that sense, we are similar to Streamlit. Plotly Dash is more of a complete die-hard coding library. Plotly Dash has an enterprise license.

Note that none of the tools above have the equivalent of Taipy Designer or Pipeline/Scenario Orchestration.

Q2) What tools/libraries are you using for the back end? Are you involved with data pipelines?

Answers: Tools like AirFlow or Prefect can be mentioned as alternatives, but these pipeline orchestration platforms are likely to be used by other teams: the DevOps or MLOps teams. Taipy Core is targeting application-based pipelines, not DevOps/Execution pipelines.

Key differences:

- Taipy Core provides execution registry, ie every single run (ie scenario) is stored in a repository, ready to be re-examined (checking which input/output was used for a given scenario), compared with other scenarios, etc.
- Taipy Core executions are usually triggered by end-users, they are unlikely to be run as a batch.
- Taipy provides visual elements to explore/analyze all the scenarios generated.
- Taipy provides pipeline versioning: if an existing pipeline gets modified (new data source, new task, etc), all the previous scenarios are typically lost. Taipy provides version management to ensure a proper life cycle for a pipeline.
- It can make sense to have a given task inside a Taipy application pipeline to call an Airflow or Prefect pipeline. This is for instance, how Taipy integrates with Dataiku pipelines.

Demonstrate the Taipy Core demo with Caching, Parallel execution, and Scenario Management.

Q3) Are you using any Data platform?

You should check if they are currently using a specific Data Platform, such as *Databricks*, *SageMaker* on AWS, or *Colab* on GCP. There are 2 articles available to integrate with [Databricks](#) and Dataiku. These demonstrate how easily Taipy integrates with Data Platforms. On the website, refer to the integration examples under Tutorials.

In addition, check if they develop using an IDE (like Visual Studio Code) or using notebooks (Jupyter or notebooks from an existing data platform).

Ask if she/he is involved with data pipelines and orchestration. If this is of high interest, spend more time during the demo session on the pipeline orchestration side. Otherwise, spend only 5 mins to demonstrate how easy it is to use pipelines and scenario management.

The typical set of slides/demos needs to be presented (from the sales kit repository).

## Selling to the Python Developer

This persona is more involved with Projects rather than pilots. He is more interested in the overall software with all its components: engines, GUI, back-end, Data integration, and deployment.

It is similar to selling to the data scientists with more focus on:

- Pipeline & Scenario Management
- Pipeline Version Management, Development, Test and Production environments, etc.
- Deployment

## Questions to ask – On top of the ones for the Data Scientist

Q1) What kind of projects are you involved in? Is it pure Dashboarding? Do you have ML/AI Engines to integrate with?

Q2) How do you deploy your project? Try to check if they use a given Data Platform  
Answer: If they use a given platform, Taipy is great at connecting to these platforms. Typically, you can develop the Full-GUI in Taipy externally and have Taipy core call the engines developed on the Data platform. We have several articles demonstrating how to integrate with Databricks, and Dataiku. It will be very similar to other platforms. [<Provide links here>](#)



## Selling to the BI Analyst / Head of BI unit

It may be very hard to replace PowerBI/PowerApp inside a given organization, especially if PowerBI has been heavily used for years. Furthermore, PowerBI is usually part of a larger Microsoft ecosystem that includes PowerApps, etc.

Instead, Taipy shines with organizations that have PowerBI but also have some staff with some Python experience. Think of a young recruit with Python skills who has to use Power BI. This will not be his/her favorite path. But now, with Taipy Designer, he will be much more at ease.

When facing such prospect, it is important to very quickly introduce Taipy Designer. In effect, Taipy Designer Drag&Drop template looks similar to PowerBI's.

We feel that Taipy can live side-by-side with PowerBI since:

- Standard dashboarding and certain applications are best implemented using PowerBI
- Other applications (highly interactive, custom GUI, complex backend) will be easily done in Taipy. Not doing it in Taipy will mean tricky/complex development, low end-user acceptance, etc.

# Competition with Streamlit

## Taipy is more performant

### On refresh

Taipy refreshes faster the various graphical widgets. Streamlit refreshes the whole graphical component.

See the map demo in the Saleskit repo.

### On the support of large datasets

For charts (line charts, scatter plots, 2D, 3D, etc) Taipy provides the decimator property that enables Taipy to manage plots with large datasets with ease, while Streamlit will collapse under such load.

See the 1M point demo in the Saleskit repo.

### Taipy supports async calls

Taipy supports Async and Sync function calls. For instance, if from the graphical interface you need to call a Python function (i.e. by the click of a button) that takes 10 minutes to execute, then you will get stuck in the Streamlit interface waiting for the job to complete. In Taipy, you have the option to call this function as a 'long job' (i.e. asynchronously), and you can move around your GUI while the job executes.

< See demo from the sales kit >

## Taipy provides MultiUser with broadcasting

Both Taipy and Streamlit manage multi-user where each user has a separate state to avoid unwanted interactions. However, Taipy provides a much richer environment by allowing data broadcasting across users. This can be very useful in situations such as:

- A global data source broadcasting to many users,
- A master user broadcasting data to his team (other users)
- Etc.

<Florian J. to provide more information on this topic>

## Taipy integrates natively with Plotly (not streamlit)

Plotly is THE gold standard when it comes to Python graphical components.

Taipy provides tight integration with Plotly. It can include any fig from Plotly inside the Taipy *chart* object.

<show piece of code>

This tight integration allows Taipy to capture events from the Plotly chart and react with custom behavior. Streamlit can't capture any event from a plotly graph.

See the two Taipy Plotly Integration demos in the Saleskit repo:

- Demo showing the plotly event capture
- Demo showing a brand new chart (financial chart) generated with Plotly and integrated into Taipy.

## Taipy provides full backend capabilities

Taipy provides full backend capabilities with:

- Data Nodes,
- Tasks
- Full pipeline orchestration with Scenario Management.
- 

Streamlit doesn't. Streamlit is purely a front-end tool.

## Taipy has Taipy Designer

Taipy Designer provides a unique tool in the Python world. The only other software that comes close to it is a software called *Anvil*. *Anvil* is really complete as a designer, but it is not really open source, has no community behind it, is not so easy to use, and is just very focused on its designer, etc. Taipy Designer is released at the end of March 2024.

## Taipy has a rich set of APIs

Taipy is the only tool that provides different options for GUI development, which should suit different preferences of the Python developer.

- An augmented markdown API  
<introduce a code snippet>

- A pure Python API  
<introduce a code snippet>
- Taipy Designer

With Taipy Augmented Markdown, you get the benefits of Markdown in terms of text formatting, headers, easy positioning, etc.

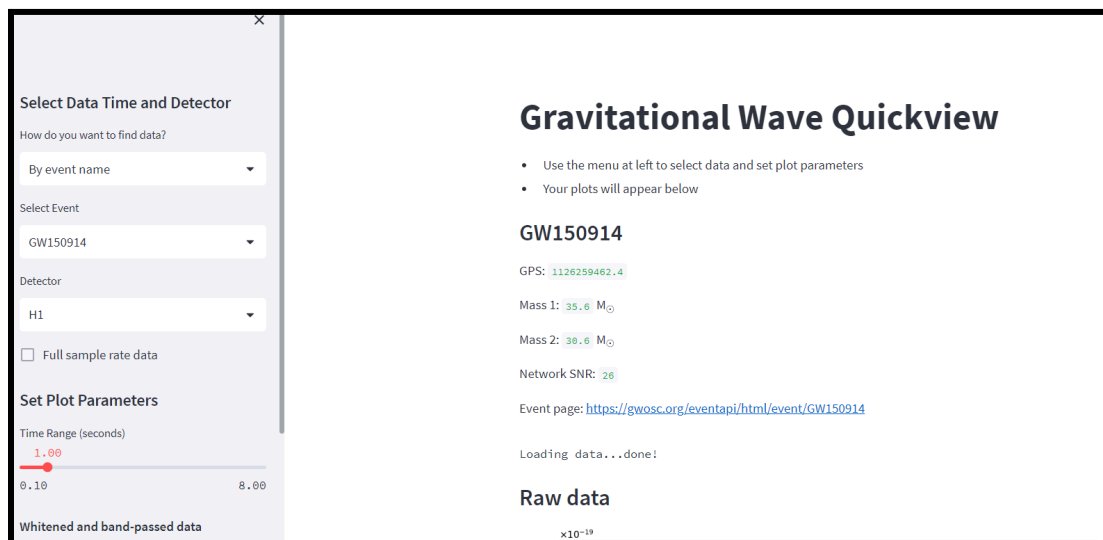
With Taipy pure Python, it is easy to manage very dynamic graphical interfaces: you can use Python loops to build your GUI.

With Taipy Designer, you get the low-code aspect, accurate design and layout, etc.

## Taipy is customisable

A Taipy developer can easily modify and customize an application's look and feel without any CSS knowledge. This is either really difficult or impossible in Streamlit.

For instance, the layout of the Streamlit screen is limited to two panels: a side panel with selectors/parameters and the main panel in the center.



## Where does Streamlit win?

### Streamlit has a larger community.

Streamlit was created 4 years before Taipy! We are catching up. However:

- more apps/examples have been developed by the Streamlit community.
- More environments have integrated Streamlit (most data platforms, etc)

### Streamlit is more connected to other graphical widget libraries.

In Taipy release 3.1, Taipy provides two new APIs to connect with 3rd party libraries.

- A tight connection with Plotly (see point [Taipy integrates better with Plotly](#))
- A new API to connect with any third-party libraries generating html code. This is valid for Matplotlib, etc.

### Streamlit is free

Streamlit is indeed free, as is Taipy. However, most corporate accounts will absolutely want to opt for the Enterprise license whenever the software needs to go into production, so this is a substantial advantage.

### Streamlit is backed by Snowflake

Snowflake is a hugely successful company that provides a leading data platform. Streamlit is a way for them to lure customers into their Snowflake ecosystem. This can be an advantage for existing Snowflake users, but this is certainly not the case with the other customers.

You should highlight the fact that we are extremely compatible with any data platform, such as Databricks, GCP/Colab, AWS/SageMaker, Dataiku, and even Snowflake!

### Streamlit is not really running on notebooks

There is no natural way to use Streamlit as a standard Python package inside notebooks. There are 2 workarounds proposed, both are more like hacks than anything satisfactory.

- Workaround 1:
  1. inside a jupyter notebook, use magic command to create your streamlit file inside a cell!

```
# This program prints Hello, world!  
%%writefile app.py  
  
import streamlit as st  
  
st.write('Hello, world!')
```

2. Then do a system call to run it.

```
!streamlit run app.py
```

This makes ou wonder why is the point of using notebooks!!

- Workaround 2: use a non-supported external package called streamlit jupyter.

# Competition with PowerBi/PowerApps and other BI tools (Tableau)

When comparing Taipy versus PowerBI, we actually compare Taipy Designer versus PowerBI. Taipy Designer provides a similar interface to traditional BI tools with its drag/drop capabilities. So BI users will feel at ease with Taipy Designer.

Main points:

- Often, BI Dpts will have some Python skills and they can see the advantage of using Taipy versus PowerBI. A young Python developer is likely to love Taipy Designer (as those guys have to use PowerBI, but they feel trapped in the tool, proprietary, old technology, etc).
- Another aspect is that often when using PowerApp/BI, you need to use its cloud version, which forces the developers to replicate its data sources in the cloud. All of this is painful.
- Customizing PowerBI/PowerApp is not easy and rigid. Taipy Designer and Taipy in general has a much wider range of customization.

# Demos

There is a whole set of demos from the Taipy sales kit Repository.

These demos are usually simple ones. During the presentation, it is worth showing a more complete demo, such as the “steel trading demo.”

Taipy Designer should always be demonstrated. So make sure you keep enough time to demonstrate Taipy Designer.