**1. Concept**

Within Guided Workflow, there are several tasks that take part in the process of creating or maintaining a workflow. Some of those tasks are:

1. Creating an excel file, along with SMEs, with the Q&As of a workflow.
2. Maintaining workflows by applying changes that are received through our SIM intake.
3. Maintaining the integrity and consistency of the Q&A files and live workflows.

After analysis, a couple of processes were automated by creating small programs for the Specialists to use whenever one of these tasks required attention. Paramount migration brought the development of new programs for daily use and efficiency as well, but also shed light on a current problem for these programs/tools:

1. **Maintenance**. Having multiple desktop applications brings difficulties when attempting to provide maintenance to a specific program, specifically if one of the programs has dependencies from another.
2. **Admin Rights**. Security measures have increased in our laptops which blocks completely any unauthorized external install into the computer, which also puts a blocker for any new program/tool to get installed or updated.

Kernel brings the concept of building these tools, as applications or functionalities clubbed together in a main web site, allowing the Specialist to have a main “Home” to find all tools needed for daily tasks.

**2. Proposal**

We propose to develop a web site which will be home for the developed applications. A scalable web-site to extend on future development of tools and that can connect with databases created within the team for better use of applications and data.

**3. Why GWF Kernel?**

Within GWF Team and PRO, there have been multiple changes, for which the team had to adapt to meet expectations and goals. Automation to deliver better and faster results, security policies update to have improved security on our laptops.

1. **Automation.** We took constant and repetitive tasks within the team and created separate tools to automate the process and deliver more accurate and efficient results. This led to having 3 different tools on the team. Some of them were not even installed in all of the Specialists PC’s, due to issues in file sharing. In a nutshell, the desktop applications were becoming time consuming to maintain.
2. **Security Policies.** Security was updated in our laptops, limiting the files or programs a user can install, to only the files allowed within Software Center. This blocked future update installations for the developed tools.

**Kernel** brings a solution to these problems and propose a scalable web application for future tools needed by the time.

By setting web-based applications we manage to:

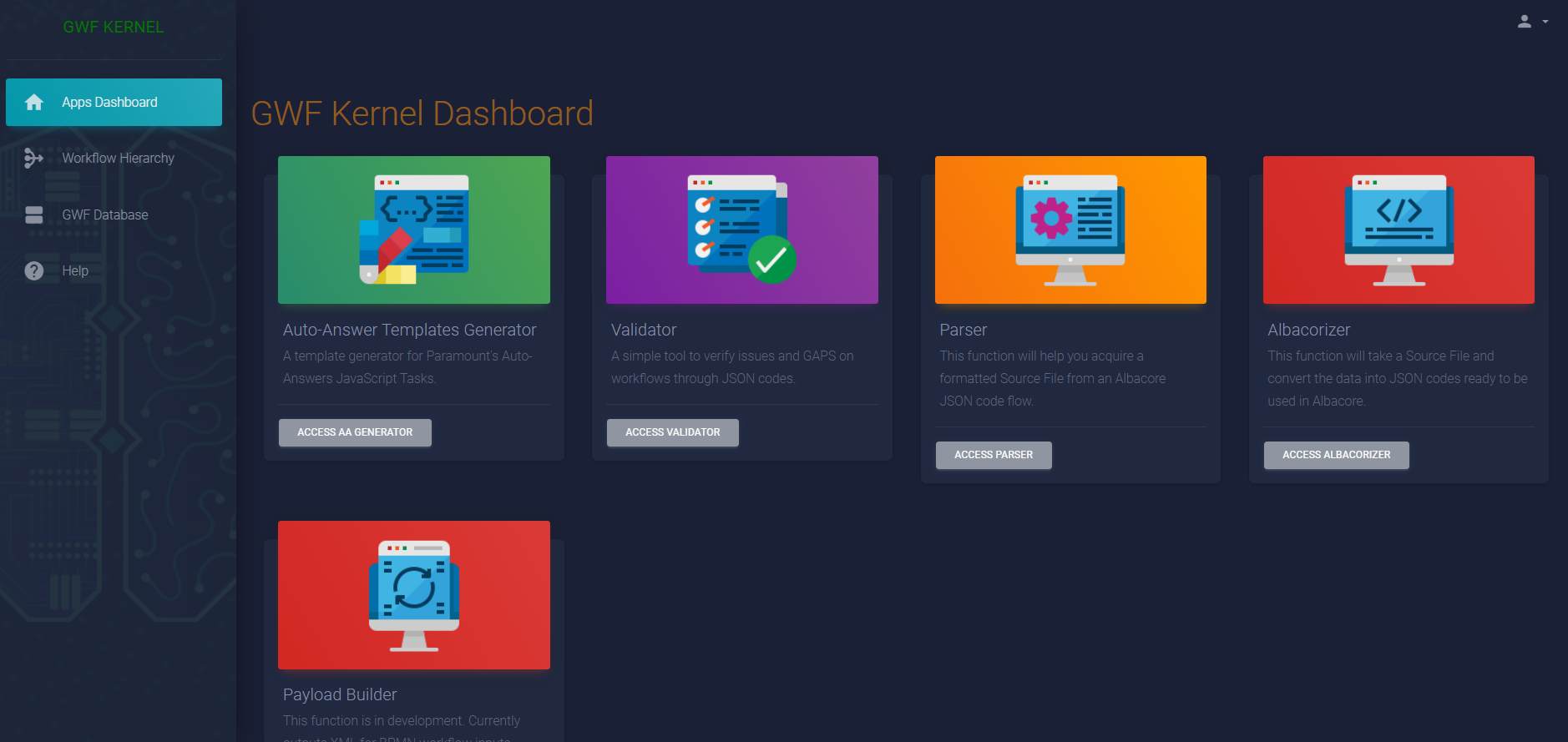
1. Control updates without affecting user. Updates would be made on the back-end.
2. No permissions would be required for installation on updates. This means maintenance to developed tools will be performed and updated on server files, so when a new tool is updated or installed, the user will just see it in the Home screen the next time he/she logs-in.
3. Web-based offers scalability and ease of implementation. This means creating new tools for the team to improve in daily tasks.
4. As we continue to optimize and improve our workflows, internal processes need as well to be automated and simplified to keep up with the continuous evolving of the team and the service it provides.

**4. Developing**

**Kernel** it’s at an advance stage of its development. We are currently at the stage of researching for a way to deploy live to a web-server for a beta test, to allow bug detection and UX (user experience) feedback.

There are several key points for developing that have already been completed:

1. Back-end infrastructure for tools and application management.
2. Simple database for Albacore Auto-Answer IDs to cover for **Auto Answer Template Generator** feature.
3. A GitHub project, for source control and back-up management.



Above Image. GWF Kernel’s Home page/Application Dashboard

Here is a short list of applications working currently on **Kernel**:

1. **Albacorizer:** Takes Source Files(.xlsx files / Flows) and convert them into two separate JSON format files, one for sections of the flow and one for questions.
2. **Parser:** It automatically access Albacore, takes the JSON codes for the specific flows provided by user and then creates Source Files(.xlsx / Flows) out of it.
3. **Validator:** It asks for the JSON codes that an admin inputs from albacore and then it validates for all the errors and the issues that could break a workflow and then print the output in a notepad local on the computer for the admin to take a look at it.
4. **Auto Answer Template Generator.** From a selection of inputs provided by the user, it outputs a template to set-up Paramount’s development tasks for the Albacore auto-answer set-up.
5. **Task Extractor.** This new addition handles copying a complex task form Paramount’s Workflow Designer and provides the Workflow Specialist with a code to implement the copied task into a different workflow container.

**4.1. Developing Tools**

For **Kernel’s** development, we implemented several technologies to accomplish the ideas for automation and to create an environment that would allow growth with little to no hassle.

* Given the distance and time-zone differences between developers, a Git hub project was configured for source/version control and to keep a back-up of the project files. This is the link for the [GitHub](https://github.com/MarcoOrtiz90/gwf_hub_repo) project for review if desired.
* Main tools were initially developed in Python. Migrating to Web represented a challenge since they work with different languages. To bridge that gap, **Django** framework was used to combine python and web technologies such as JavaScript.
* **IDEs**. The **I**ntegrated **D**eveloping **E**nvironments used for building the back-end with Django were **PyCharm** & **VSCode.**

**4.2. Deployment Challenges and Requirements**

**Kernel** is currently at a stage of its developing in which deployment is necessary to test functionalities and to support GWF Team with upcoming tasks.

A web service within Amazon’s VPN would be a suggested solution for this. This will allow a dedicated URL for the site so the Specialists are able to access when needed. After presenting the idea to a POC SDE for guidance on next steps, an AWS account would be the initial step in the path for deployment Kernel but we are yet to decide which AWS service will be the appropriate for Kernel’s needs and features.

For Amazon members and Teams under certain criteria, there is availability for an Internal AWS account.

**4.3. AWS Account Creation.**

The Following information is a compilation taken from Fleet Management’s [Request Internal AWS Account](https://fleet-management-console.amazon.com/aws_account_requests/new) and [Internal AWS Accounts SOP](https://w.amazon.com/bin/view/MAWS/Internal_AWS_Accounts)

Any internal request for an AWS account goes through an approval chain. In the above mentioned paged, there is a form to **request the approval of an AWS internal account**. As an important note, a Bindle is needed to manage permissions on the account.

* **Bindle.** Bindles is a new ownership and permissions management system. A Bindle is a team-owned collection of related resources for which permissions should be managed as a single entity. Visit Bindles [SOP](https://w.amazon.com/index.php/Bindles) for more details, or [Home](https://bindles.amazon.com/software_app/new) page for details on creating a Bindle.

**4.3.1. Steps to Account creation**

1. **Consult your Team**. While this document describes the general steps for acquiring an AWS account for internal use, each team will have adopted its own conventions for how this applies to them. Check your team's internal documentation to be sure you're going about this the right way.
2. **Create an account email list**. [Create an email list](https://mail-admin.amazon.com/cgi-bin/email-list.cgi) for your team if you haven’t already done so.
   1. General guidance is to have an AWS account per stage of your service. More details are available in the [Account FAQ](https://w.amazon.com/bin/view/MAWS/AccountsAndBilling#How_many_accounts_do_I_need.3FAWS).
   2. An email alias of the form <team>+<service>-<dev|beta|qa|prod>@amazon.com also works, and allows use of a single team email majordomo list to manage multiple AWS accounts. This removes the need to create more than one email list and pre-existing team aliases can also be used (assuming subscription is correctly locked down as described below). This only works with a Majordomo email list. This will not work with a Mail Enabled ANT Group (MEG).
   3. Email address with above format ( plus followed by dash e.g. actual-team-email+service-dev@amazon.com doesn't work when registering the account with Isengard. Try using just the plus signs. E.g. [actual-team-email+service+dev@amazon.com](mailto:actual-team-email+service+dev@amazon.com).
   4. Be sure to check the following option on your email list:
      1. Subscription policy: closed
      2. Block external mail: no
3. **Request an Internal AWS account**. Subscribe to the [access-maws-announcement@](https://email-list.amazon.com/email-list/email-list.mhtml?action=search&name=access-maws-announcement&x=0&y=0) email list. Include your team alias as well as the account alias you created above. This is a moderated list that will only be used by the ATS team to make announcements regarding AWS accounts. For questions and discussions on internal Accounts, please subscribe to [maws-interest@](https://email-list.amazon.com/email-list/expand-list/maws-interest)
   1. If the subscribe page does not load [too many members?], you may have to hit the rest API directly:  <https://email-list.amazon.com/email-list/email-list.mhtml?name=access-maws-announcement&subscriber=><YOUR TEAM OR USER NAME HERE>&action=subscribe
   2. Go to the [Fleet Management Console](https://fleet-management-console.amazon.com/aws_account_requests/new) to request a new AWS account.
      1. AWS Accounts cannot be associated with a "Personal Bindle". Please use a Bindle owned by a Team.
4. **Create AWS IAM Users.** Once account has been created, AWS users need to be created who will use the actual services. You should use IAM Roles for this. Check [Conduit’s IAM Guide](https://w.amazon.com/bin/view/Conduit/IAM_Guide/) for recommendations on how to access AWS.

**AWS accounts** approval times vary depending on approvers.  Approvers can be reached out to ensure getting approval in a timely fashion**.** Once approved, the account creation takes about 10 min and an email is sent to the requester upon successful creation of the AWS account. Status request can be tracked [here](https://fleet-management-console.amazon.com/requests/?tab=aws_account&default=TRUE). Learn more about approval process [here](https://w.amazon.com/bin/view/FleetManagement/PAO/Projects/AwsAccounts#FAQ).

A previous request for an AWS account was made for the intention of hosting the tool but it was rejected.

Refer to this [link](https://fleet-management-console.amazon.com/aws_account_requests/235819) for details. The suspected reason is that the previous attempt was made through a personal Bindle and the AWS account was requested as individual and not as a team account.

**4.4. Ongoing Development**

It is expected that **Kernel** will be live to support GWF Team. At the moment, the project is unable to move forward due to the lack of a domain to host the web application. None the less, **Kernel** continues to centralize new applications. With the ongoing Paramount Migration Project, new applications have been developed and added to the tool and more are expected to come.

**5. FAQ**

**What other tools does it interact with?**

Only to Albacore and if a user is currently authenticated.

**How do you authenticate for use?**

The tool does not need authentication for use. The tool sends a request for a link that the user will be able to access to if it has been previously authenticated from the company’s user credentials.

If the user is not logged in to the web app that the tool is requesting, company authentication will be prompted.

**Could the tool make modifications to any other software in Amazon?**

No. All functionalities request information from the previously mentioned apps (desktop/web) but it does not have functionalities to make changes to a software used in the company.

**What are the software packages needed for developing?**

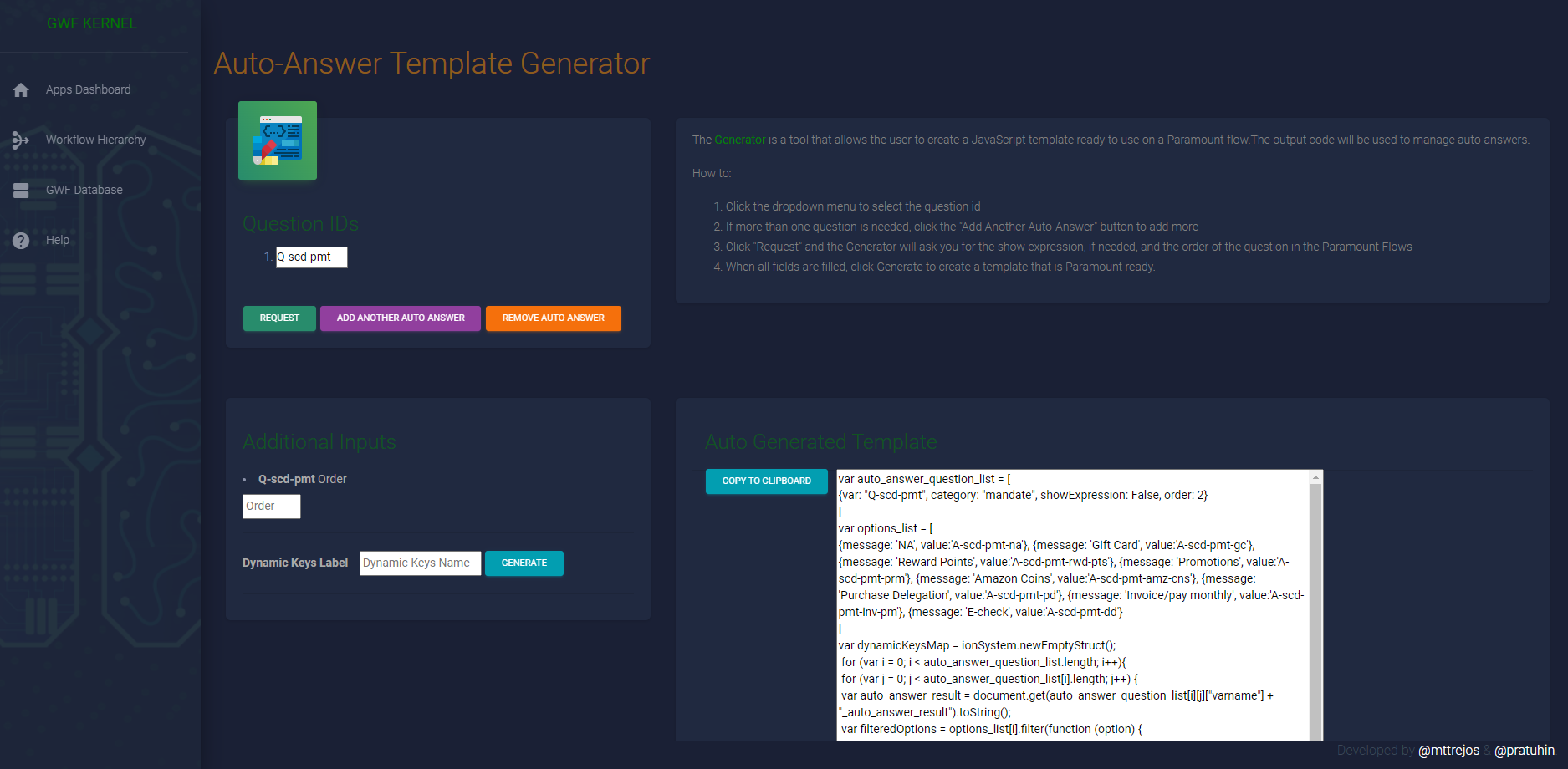
Languages: Python, JavaScript, HTML, CSS

Python based packages:

* Django

JavaScript based packages: NPM needed (Node Package Manager). \*\***Admin Rights are required to install below packages**\*\*

* React.js



Above Image. Kernel’s Auto Answer Template Generator. **Functionality created for Paramount**.