powered by SPG DATA SOLUTIONS

AI-supported Code Migration of Data Analysis Scripts (SAS/Python)

A Case Study from Banking using LegacyLift

ADV Data Excellence Konferenz
Wien, April 2024

LegacyLift

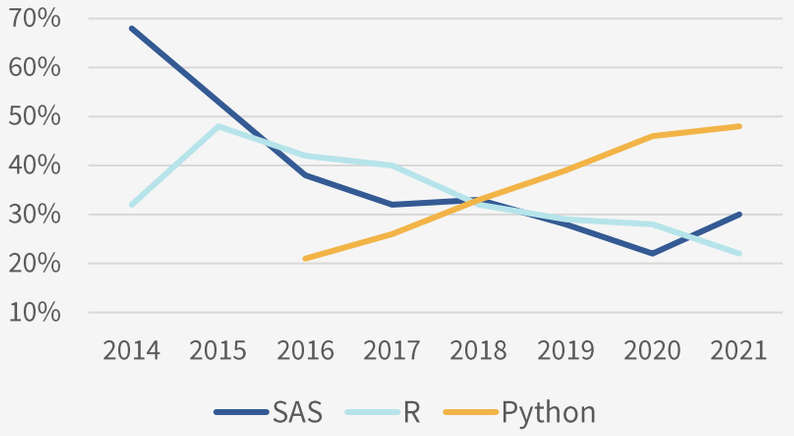
LegacyLift

AI & Legacy Code Migration

SAS-Scripts to Python-Scripts?

SAS loses its attractiveness by younger professionals, while Python has gained popularity.

SAS, R or Python: 8-Year Trend



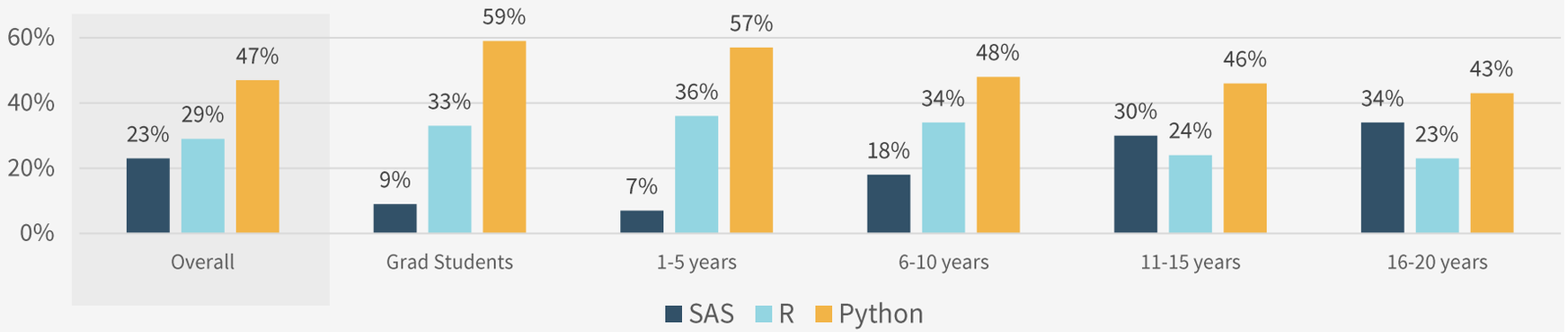
Stack Overflow Most Popular Programming Languages

For several consecutive years Python is fighting to be in TOP 3 most popular programming languages, while SAS barely makes it to the ratings. R found its place in the middle.

70%+ SAS programmers are **40+** y.o.

Students choose to learn Python **57%**

SAS, R or Python Preference by Years of Experience



Why Python?

1. Versatility
2. Fast Growing Ecosystem of Libraries and Packages
3. Community Support
4. Ease of learning
5. Cost Effectiveness
6. High Job Market Availability

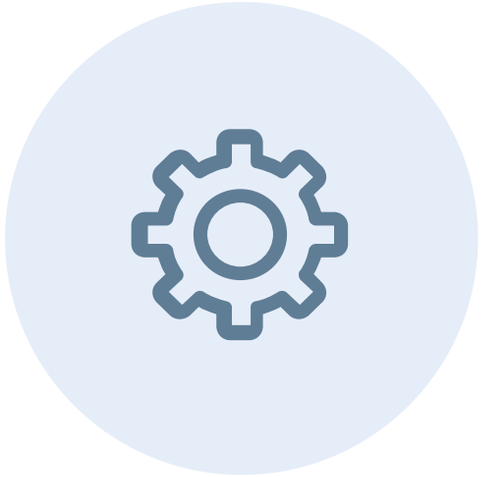
Source: BurchWorks – Industry Insights, 2020, Stack Overflow Developer Survey 2018-2023; Zippia.com, own observations

AI has the potential to provide substantial benefits when implemented in specific use cases in financial industry, according to McKinsey & others

25-40%	**Productivity Increase**	The direct impact of AI on software engineering productivity. Impact for banking sector is estimated to be the highest after high tech industry.
9-15%	**Value Potential**	As percent of operation profits for key use case, **legacy code conversion** being the biggest value driver
42%	**Employees regularly using AI tools**	While total of 83% percent of all respondents from financial services industry say they've had at least some exposure to AI tools
44%	**Reported Cost Decrease**	Across respondents in risk departments, 5% reported more than 20% cost decrease, 13% - a decrease from 20-29% and 26% a decrease of less than 10%

Source: The economic potential of generative AI: The next productivity frontier, McKinsey, 2023; The state of AI in 2023: Generative AI's breakout year, McKinsey, 2023

What to look for in a code translation and migration tool?



Ability to utilize the best AI models available

Ability to switch to a different and better AI model as soon as it becomes available, as better and more powerful models are likely to be available in the next months

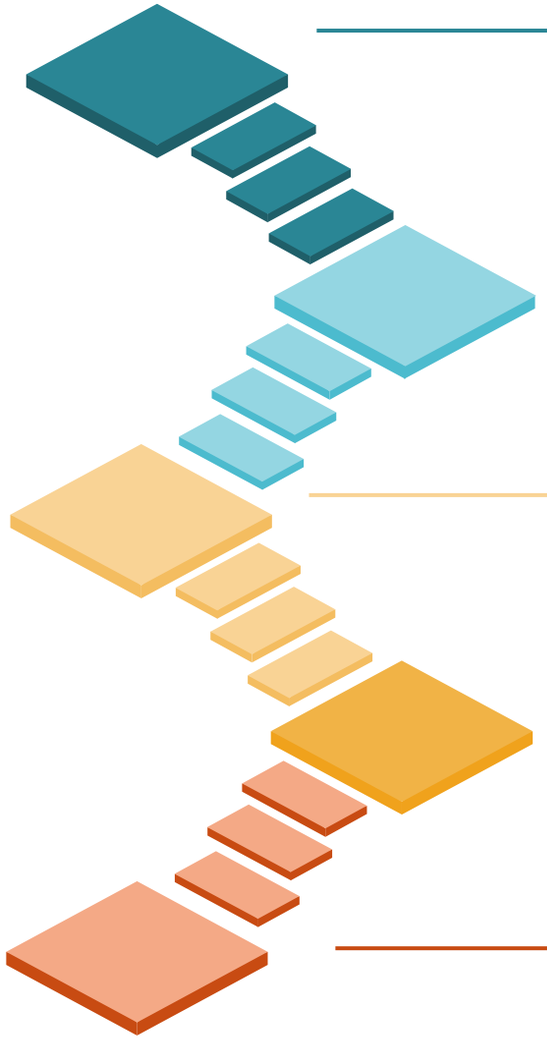
Customized for your language pairs and data architecture

RAG-pattern code database filled with examples and adjusted to match the particular target architecture, coding standards and requirements coupled with configurable prompt engineering

AI-Testing & -Documentation, Enterprise-grade Workflows

AI-automated testing and code documentation to support the process beyond the code translation. Ideally, enterprise-grade workflows and full audit trail.

Where to use AI in a code migration project?



1

AI Code & Architecture Analysis

Split the legacy code base in code complexity buckets, extract basic stats
Create architecture flows and ER diagrams

2

AI Code Commenting

Add in-line comments to legacy code
Both prior to translation and after translation

3

AI Code Translation

Translate code automatically based on the customized AI agent (target language libraries, styles, architecture examples)

4

AI Test Generation

Create automated tests
Decrease manual quality control efforts until code is production ready

5

AI Documentation

Generate human-readable summary of what each particular script/piece of code des

LegacyLift

Case Study

Bank-wide SAS-to-Python Migration

powered by SPG DATA SOLUTIONS

Case Study: How to utilize AI to migrate SAS code to Python and Snowpark

Problem: A bank wants to migrate away from SAS and into Python to modernize its reporting and business analytics. Python with Jupyter and Snowflake/Snowpark are the tools of choice.

1. Human – Written SAS Code

- Own estimate **5700** PDs from own **business SAS** users
- SAS-transformations
- Calculations work
- Analytics & Reporting

Python/Jupyter

2. Auto-Generated SAS Code

- **500.000 lines** of code
- ETL code
- Visualizations
- Monitoring Analytics

Python Snowpark

Challenges

- Very **high effort estimates** if done internally
- **Blocking** of internal resources, i.e. almost impossible to be done internally

- Which **options** are there to supercharge such a migration project with AI?
- Or is AI only good for generating images and LinkedIn posts?

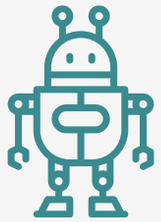
Approach:
LegacyLift + SPG + minimal bank team

PoC Results:
Coming up at the end of presentation

Case Study: How to utilize AI to migrate SAS code to Python and Snowpark

LegacyLift

- SAS → Python AI Agent
- SAS → Snowpark AI Agent
- SAS → Python AI Agent that maximizes SQL



Project Team

External Team

Project Manager
0.5 FE

Business Analyst
1 FE

Data Engineer
2 FE

Client Side

SAS Expert
0.2 FE

IT SAS person
0.2 FE

Client support is vital due to familiarity with the codebase making customization easier and resolving issues faster.

PoC Results

Human-generated SAS code

Savings:

Low Complexity:	Mid Complexity:	High Complexity:
60 %	50 %	40 %

SAS-generated ETL:

+20 % Improvement over human generated code translation

Overall project savings vs fully human code migration

50 %

Demo: Legacy Code Migration in Action using

LegacyLift

Thanks for your attention!
Questions?

Karl Ivo Sokolov
Managing Partner

ivo.sokolov@specific-group.com

Tel: +43 664 8337312

Daria Kovalchuk
Product Owner

daria.kovalchuk@specific-group.com

Tel: +43 677 64373288

Daniel N. Huber
Principal Data Scientist

daniel.huber@specific-group.com

Tel: +43 1 489 25 28

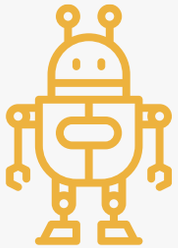
LegacyLift

Backup Slides

Information on LegacyLift

We provide a range of model choices, all optimized with carefully crafted prompts, to ensure the best performance and results

Utilize the power of AI Agents



- Choose between multiple secure public and private LLMs
- Select the language pair you need
- Use specific libraries based on your target system architecture
- Maximize or minimize SQL usage
- Adhere to any other company standard by creating new agents

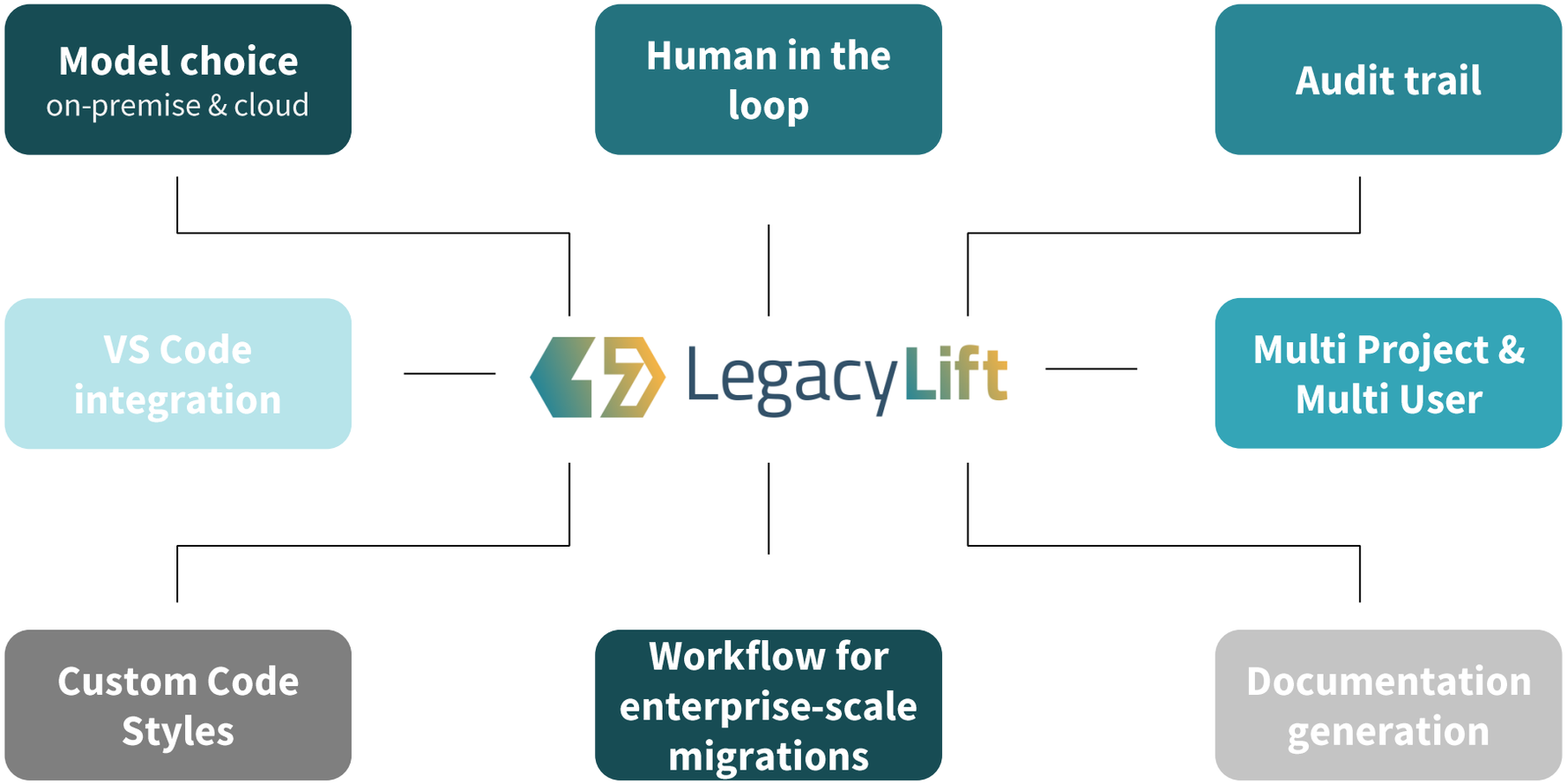
End – 2 – End Migration Workflow

- Human–in-the–loop
- Fits into your modern system architecture
- Testing & Validation
- An experienced development/consulting team at hand

Available Languages

SAS		Python / also optimized for Snowpark
COBOL	→	SQL / also optimized for Snowflake
PL/1		Java

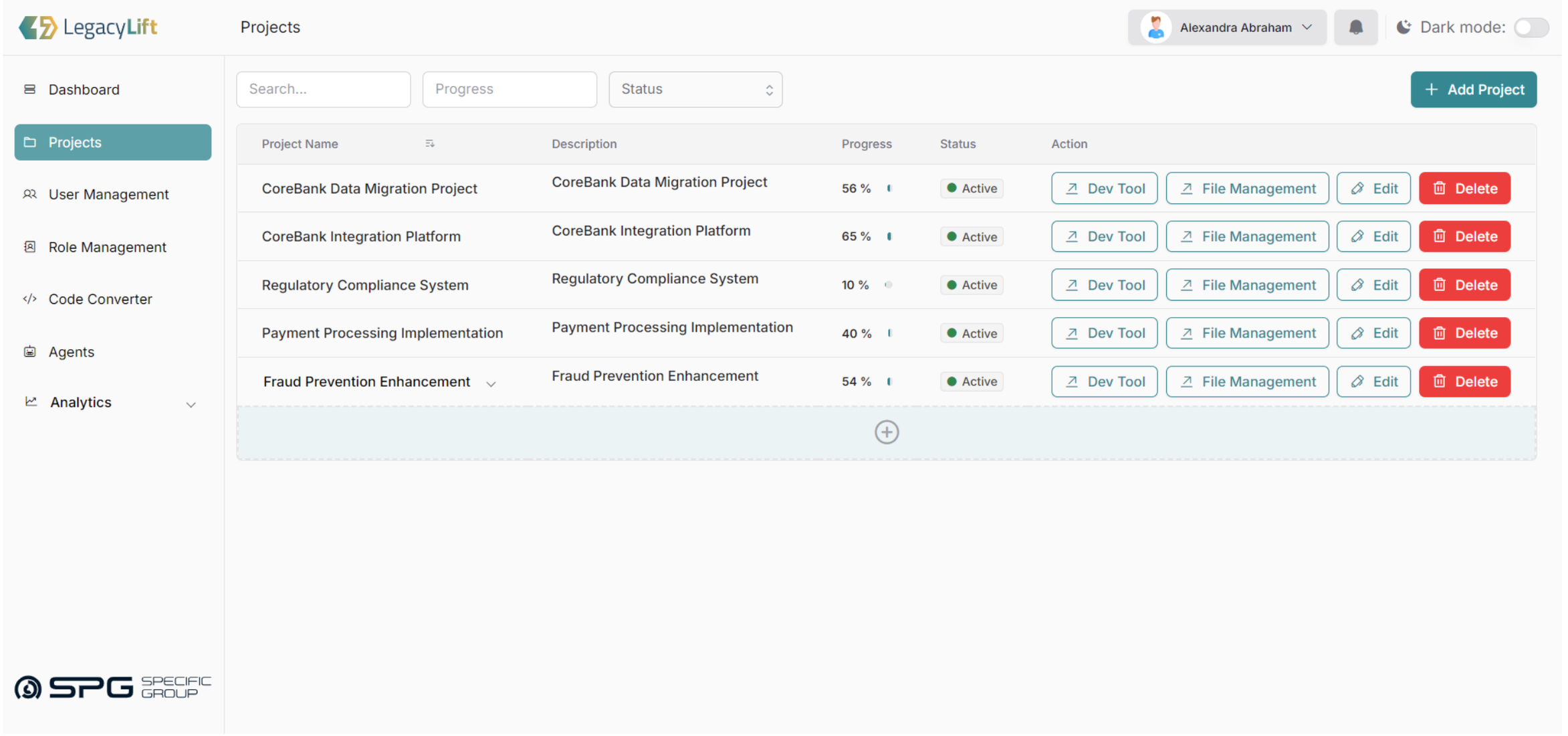
Legacy Lift offers all features needed for enterprise scale code translation projects



This it how it works:
Upload the code example for AI analysis, which splits the code into functional blocks and selects the best example from the RAG database. This optimized example, along with tailored prompts, is sent to the LLM model, with the process iteratively refined through multiple feedback loops for accuracy and efficiency.

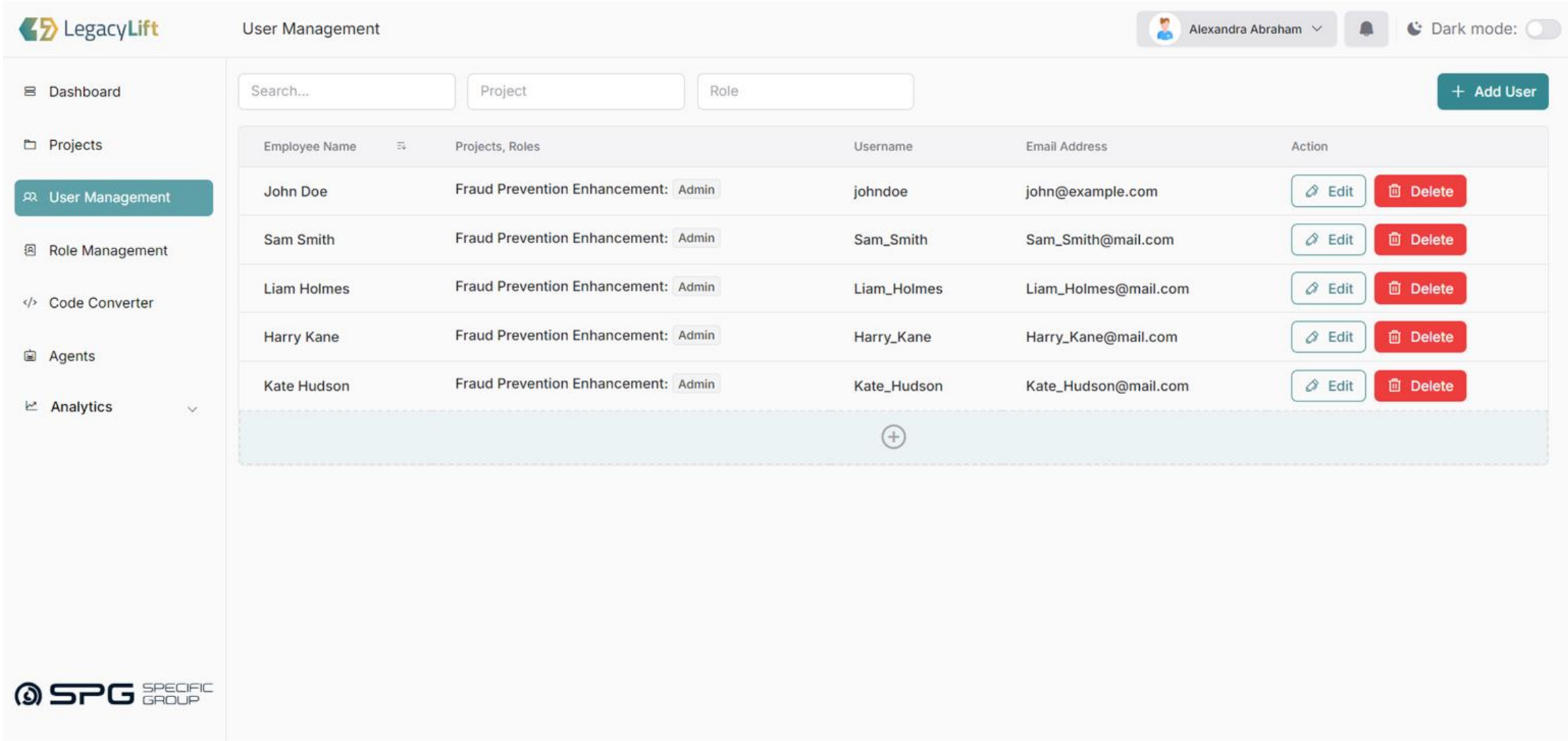
Features – Project Management

Effortlessly track and manage your translation projects

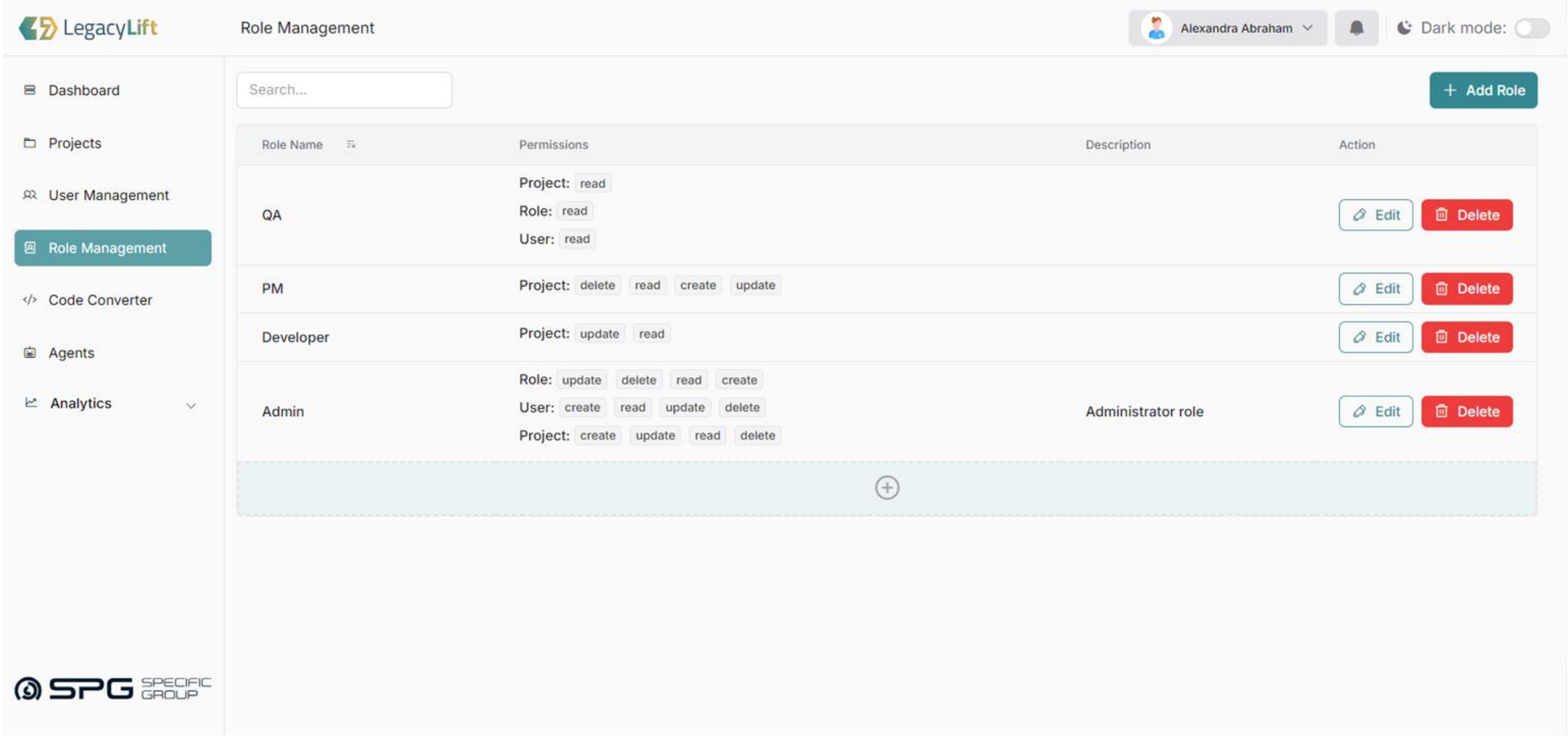


SPG SPECIFIC GROUP

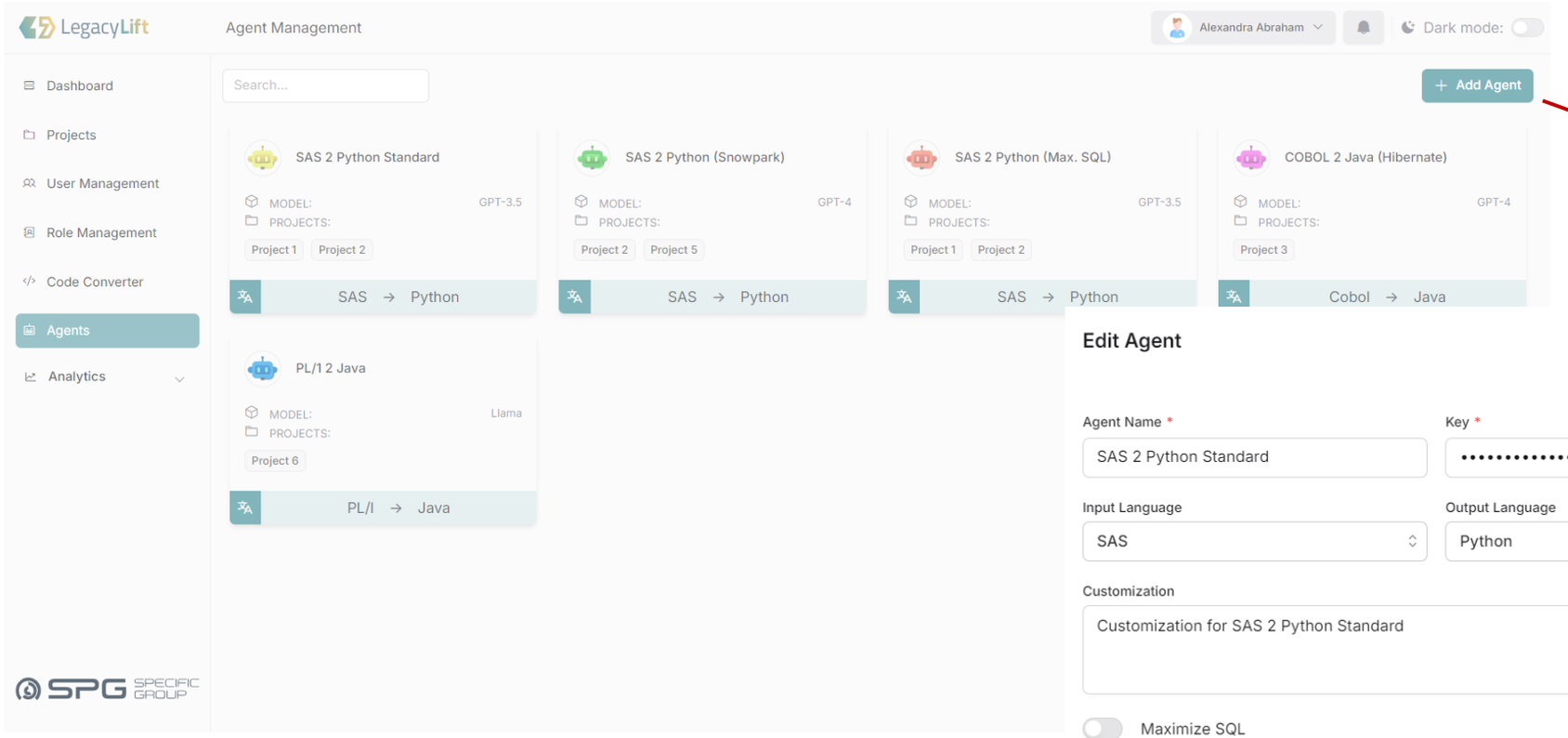
Efficiently administer user access and permissions, ensuring secure and organized process



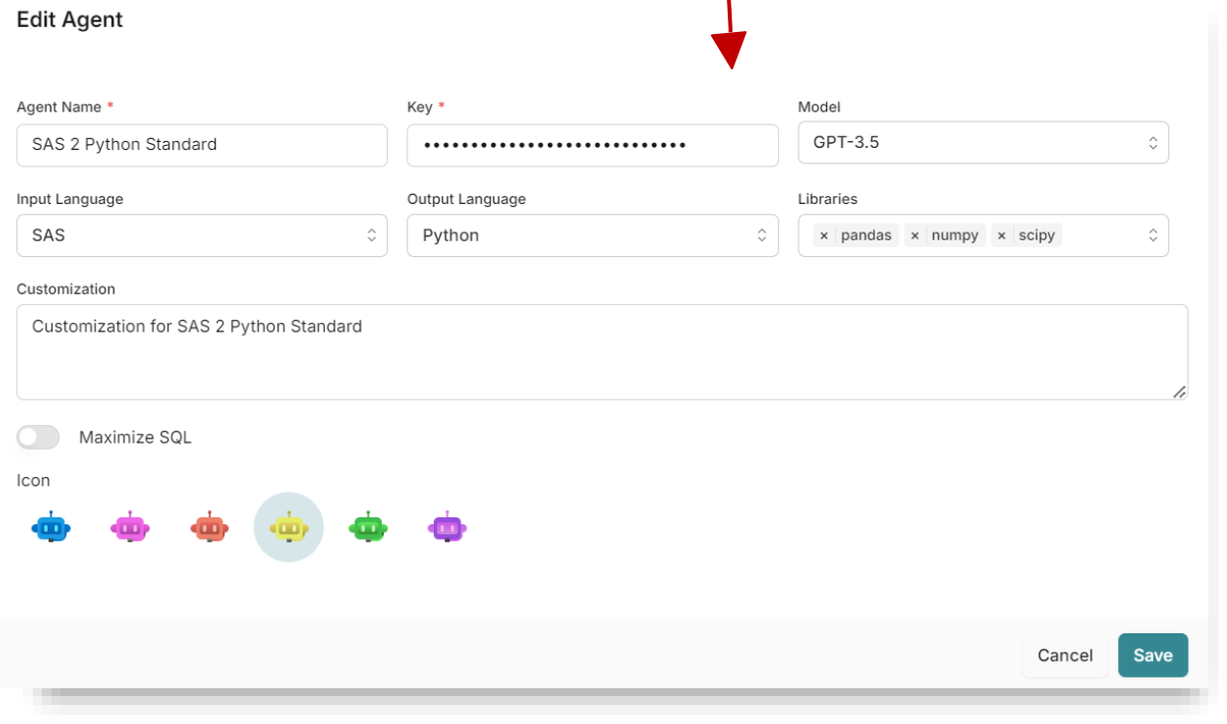
Defining access roles within the system to manage permissions and responsibilities



Easily create and adjust agents to your company requirements

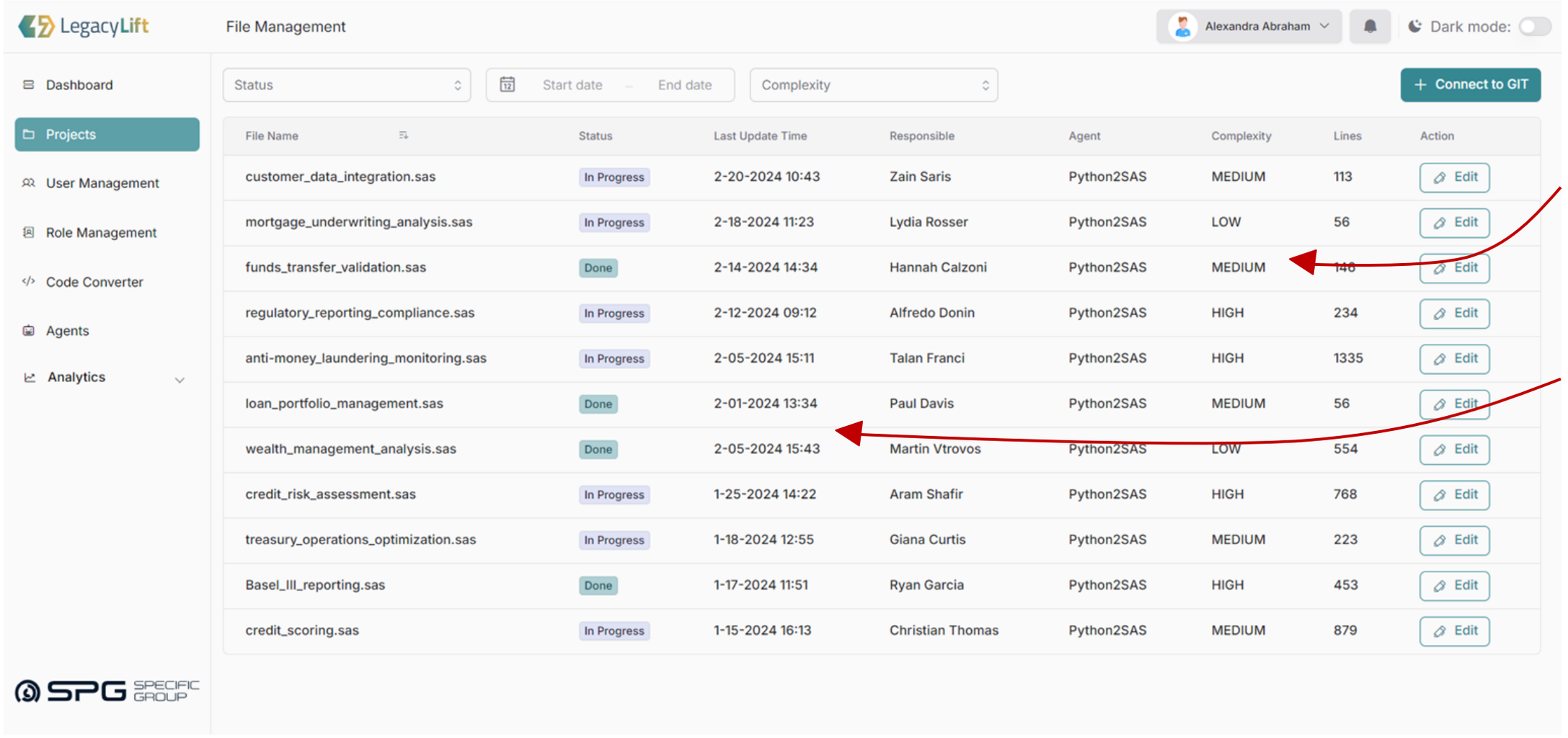


Get an overview of available agents and create your own



Select language pairs, libraries, LLM Models. Add customized prompts

Connect to GIT to see project files overview and their progress and complexity

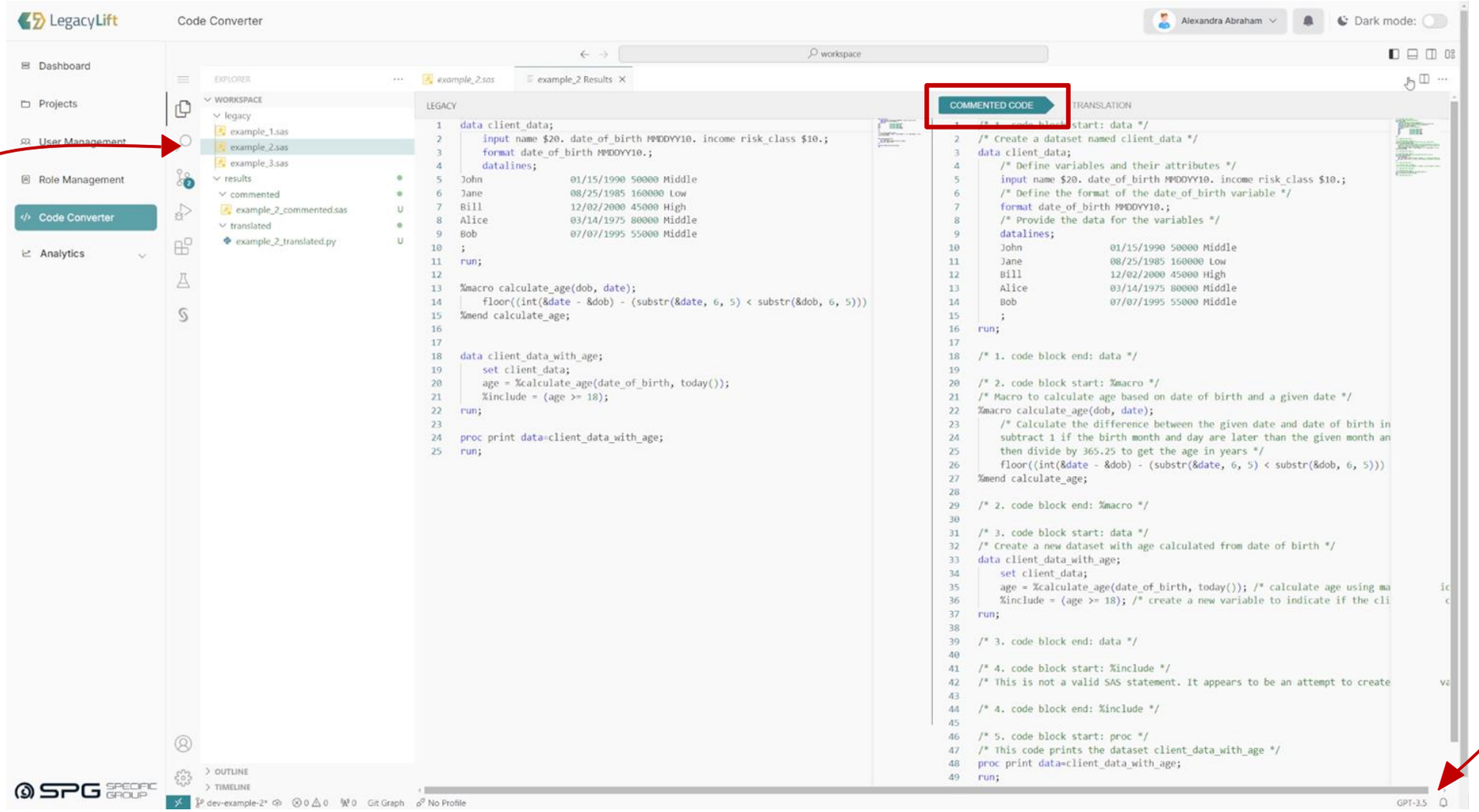


Analyze complexity with AI

Track project progress on files level

Clear and concise comments within your legacy code for easy understanding and reference

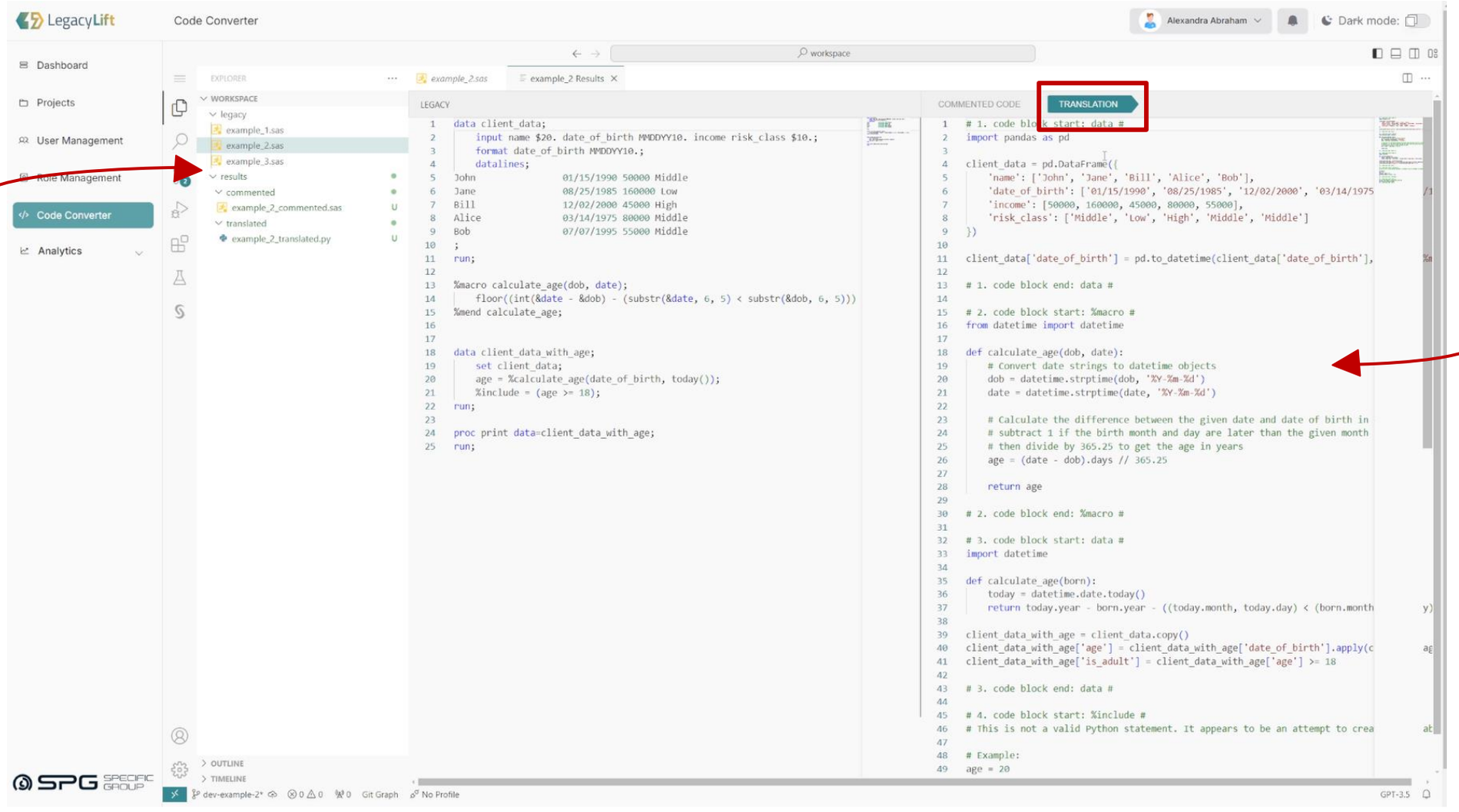
Use the entire VS Code functionality right inside Legacy Lift, such as git integration, automatic unit testing and more



Change AI Agent with few clicks

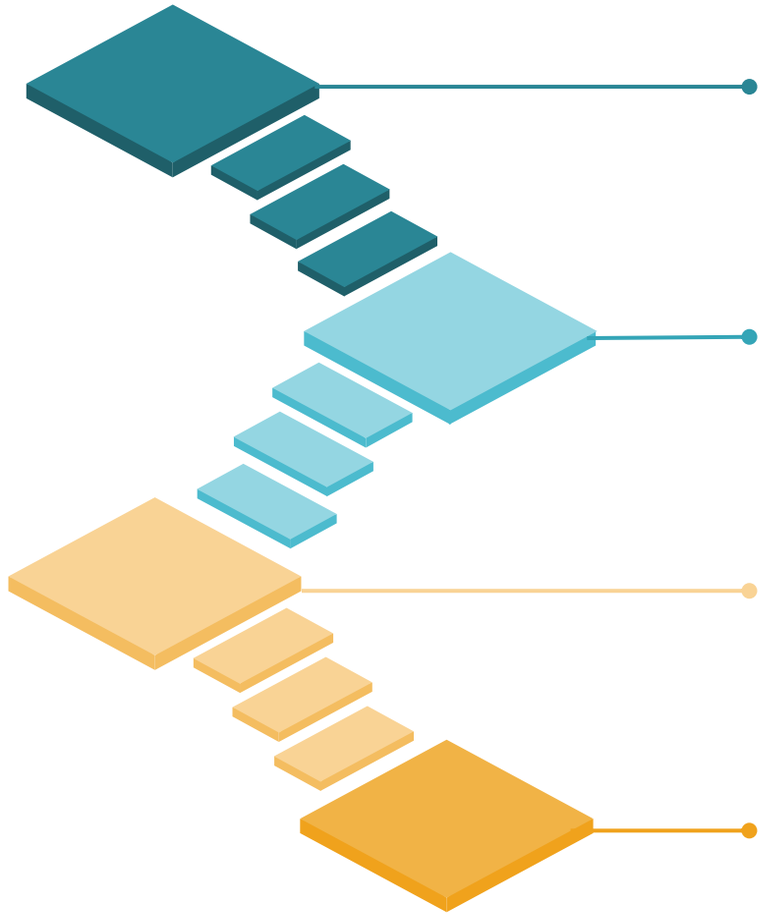
Seamlessly converted legacy code into modern languages, ensuring functionality and efficiency

Easily navigate, organize, and manage your code repository



Benefit from convenient syntax highlighting

We still need capable human teams to complete the entire migration process. Human-in-the-loop + AI agents + workflow tooling fit for the enterprise



1	**Analysis**	Client Side	Thoroughly assess the legacy codebase to understand its structure and requirements for a successful translation.
2	**Architecture**	Client & SPG Side	Strategize the translation process, outlining the plan and architectural considerations for the updated code.
3	**Development & Integration**	SPG Side	Implement the designed changes and translate the code with the help of AI tool, incorporating modern language standards, data integration.
4	**Testing & Rollout**	Client & SPG Side	Test the translated code to ensure functionality, identify and rectify any issues. AI-generated documentation.