

Tales of a MicroServices Provider

Who Are FinoComp?

- FinoComp is an award winning software development company founded in 2015 with a unique approach in developing software as independent MicroServices which are transforming the technology landscape for the Wealth Management Industry
- These MicroService components tackle regulation, reduce costs and improve the customer proposition
- Currently Employ 40~ full time staff across Australian Development Centre and UK
 - Australia: Development
 - UK: Sales, Support and Implementation Services
- All revenue currently generated from UK wealth management market



- Now owned by...



Our Vision....

- Build an eco-System of **MicroService** components to support the operation of Wealth Management organisations at all layers in the Wealth Management industry.
- For each **Component** to be deployable as an independent Software Package OR any number of our components to be deployed and operate together as a combined suite of components.
- **Automation** at all layers: deployment, testing and monitoring.
- Unrivalled **Quality** through automation.
- **Scale** beyond all possible needs.
- Truly **Global** product.
- To become the **Leader** in WealthTech Microservices technology.

Our Clients

novia /

novia /
global

SEI New ways.
New answers.®

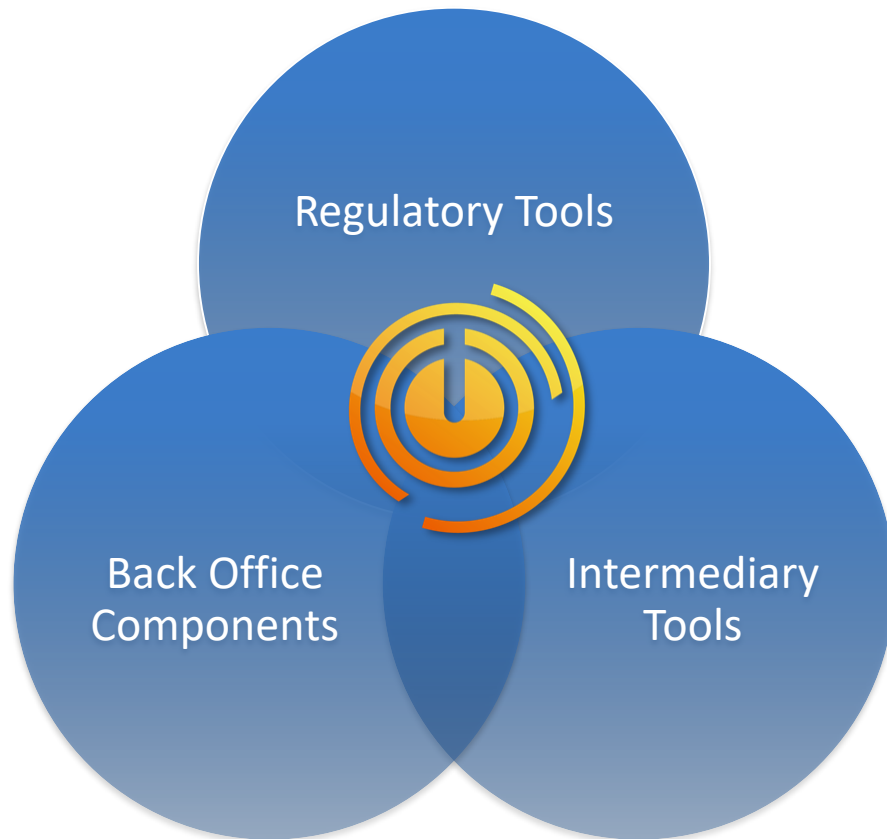


nucleus

AEGON
c·funds

 **FINOCOMP**
The Financial Services Innovation Company

Our Growing MicroService Eco-System



MicroServices... What to watch out for?

Have a Guiding Vision of the Future

- Build a view of long-term development scope
- Help in defining the boundaries of all MicroServices in your eco-system
- Prioritising the sequence of development

Don't be Afraid of Separate Database Schema

- It's important for your DB schema to be isolated and stand alone.
- To be a fully self-reliant component, it should not have a dependency on being a part of a greater schema.
- If a component requires access to other data it should be integrated using service layers or data pumps..
- You will also get massive performance benefits from distributed parallel processing.

Don't Fall Into the Trap of Microservice Boundary Creep

- Each microservice should be excellent at doing what it does. It is the software designer's responsibility to make it stay that way.
- When a change comes through that takes a component away from its intended responsibility, it may be better to step back, re-assess and perhaps create a new microservice that interacts with the existing
- It may cost slightly more and take slightly longer to implement it in an architecturally correct manner.
- The benefits of those decisions will pay dividends every day afterwards both to the development team and the clients who use the software.



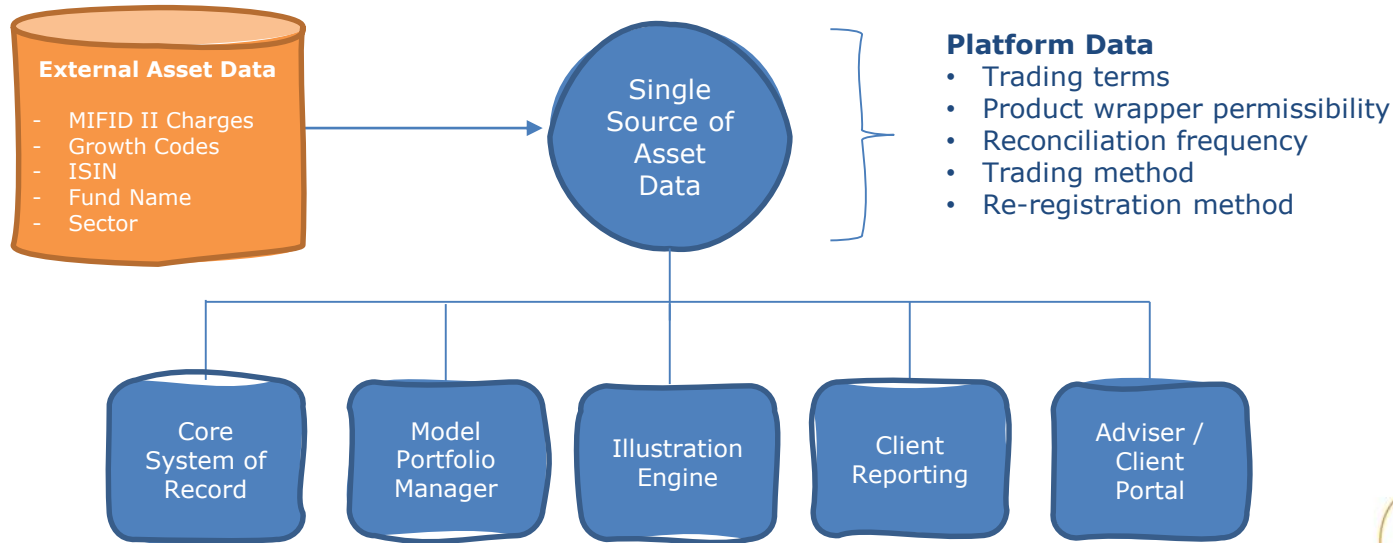
Core principle of MicroService

- The core principle of MicroService architecture..... Stick to one job and do it well!



An example MicroService....

- It's job... As the master of asset data it needs to ensure asset data is synchronised across the technology eco-system of a wealth management platform / platforms



The challenge we were presented...

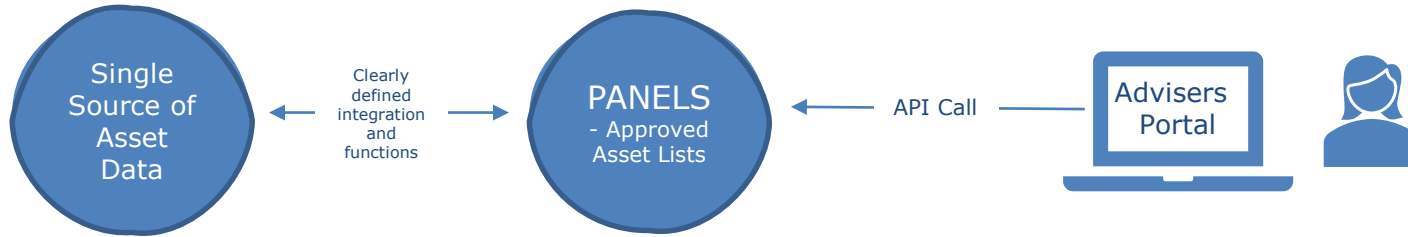
- Client requested enhancement to Single Source of Asset Database
- Preferred asset lists to be made available for advisers
 - Requirements gathering and architectural design meetings....
 - Core and fundamental change to SSAD and it's core purpose
 - It would be required to understand Intermediary Hierarchy...BOUNDARY CREEP!
 - Natural desire to build within the SSAD MicroService – Is this right?
 - Commercially (Initial build)– Would cost less
 - Timing – Would be quicker and more simple to implement
 - Long term – Would cost significantly more!
 - Tough conversation with a customer.... **It'll cost more and it'll take longer!**
- The right decision isn't always the most obvious one



The outcome...

A new MicroService developed... PANELS

- It's purpose....to maintain a list of approved asset lists for intermediaries and intermediary groups / networks



Why was this decision made? To ensure the benefits of MicroService architecture are realised?

- Easier to understand issues
- High **quality** as there is less "surface area" to test
- Better **performance** because services have their own processor so have no impact on core systems performance
- Hardware can be **scaled** just where needed to save cost
- Faster to **evolve**, so they get even better at what they do over time
- **Faster** delivery of software

The core principle of MicroService Architecture

Stick to one job and do it well



Contact

Contact details

FinoComp
No 1 Poultry
London
EC2R 8EJ

E: contact@finocomp.com

T: +44 (0) 7717 847 146

