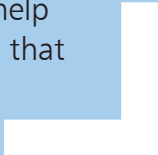


## WHITE PAPER



### ARCHITECTING FOR REALITY

This paper will explore how a metadata-led approach to understanding applications can help architects regain control over the applications that run their business.



## The Challenge For Architects

It's not easy to be an IT architect. Architects are under constant pressure to create and enforce efficient IT models for delivering business services. This is difficult enough thanks to shifting corporate strategies and business requirements, but is made still more challenging by the reality of today's IT portfolios. In particular, the complex application portfolios that automate core business operations.

Operations are run by highly-customized applications built in dozens of languages and modified over the course of decades. These applications are often poorly understood, without current documentation, and out of alignment with architectural standards. Yet the vast majority of new business requirements impact, extend, and reuse these complex existing application portfolios.

As a result, architects should be especially conscious of how to control and improve the architecture of their current application portfolios. To do so requires architects to have insight into these systems. Not an easy task when applications have grown to millions of lines of code developed in COBOL, Assembler, VB, Java, and various combinations of languages.

## An Approach To Understanding

An effective strategy to understand the architectural reality of the application portfolio is via metadata-based abstraction. That is, by 'tagging' and grouping application artifacts and their inter-relationships (e.g. data flow), we can harvest the 'as-is' architectural model of the application portfolio. When equipped with this intelligence, architects can improve control over their complex applications.

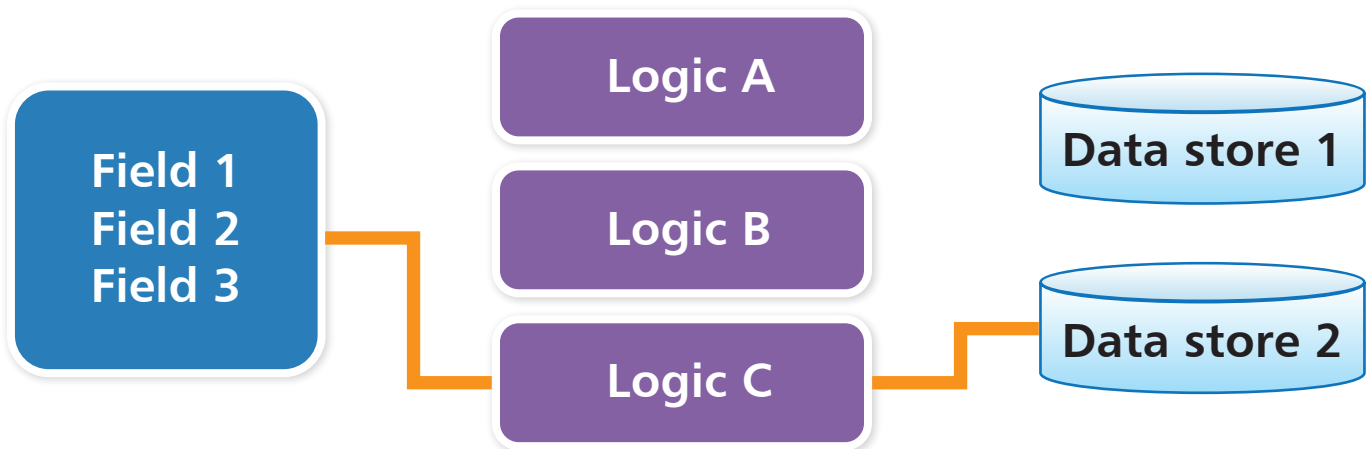
Architects can begin with a 'top-down' approach by identifying portions of the application portfolio that should be targeted first. Application Portfolio Management (APM) tools can facilitate this process to narrow the scope of analysis. Once identified, architects can then proceed to identify processes within the prioritized applications. This process is greatly facilitated by generating a foundational understanding of the targeted applications via static analysis.

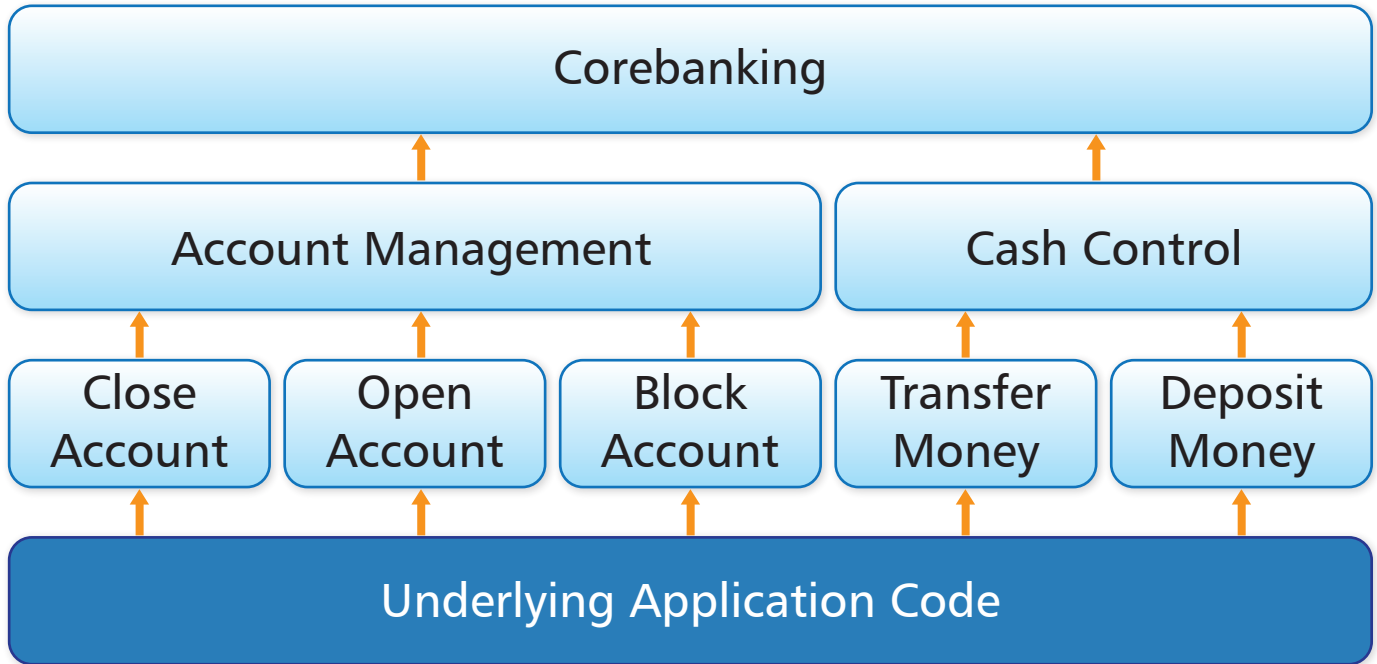
## Role Of Static Analysis

Static analysis technologies have evolved significantly in recent years. Architects can now load source code, written in numerous languages, into these technologies to understand the universe of coding artifacts under management. Without appropriate business context however, static analysis at a technical level is primarily useful for developers.

Architects can build on static analysis by marking and grouping flows as part of architectural models. For instance, we could begin by identifying a 'seed' field in an application. Let's say we start with a field in a screen labeled "Close Account". We could trace the logical flow from this field through application logic to a data store, and possibly back through other logic. This could be tagged as part of the "Close Account" business process.

Users can aggregate these process flows within an application with other related concepts. For instance, the "Close Account", "Open Account", and "Block Account" processes could be grouped as part of an "Account Management" process.





But tags do not need to be organized strictly in a hierarchical fashion. After all, our defined business processes could be reused in multiple broader processes. Further, different kinds of metadata are useful for describing and organizing the application code depending on the goals of the architect. For instance, the “Close Account” process could be described as being managed in India, by Outsourcer A, and subject to Anti-Money Laundering regulations.

Close  
Account

**Process:** Account Management

**Geography:** India

**Security Level:** High

**Managed by:** Outsourcer A

**Regulatory:** Anti-Money Laundering

Rich tagging structures like this allow architects to understand and manage their complex application portfolios through perspectives that are useful. Let’s look at how architects can use this information to better control their application portfolios.

## How This Improves Control

### Example Value 1: Understand Applications

The sheer complexity of application portfolios means that no one person can ever fully understand it. But that is not necessary when architectural models filter perspectives on the portfolio. Abstracting views based on the groupings we have defined allows users to reduce ‘noise’ in their views of the application portfolio. Irrelevant aspects of the portfolio can be abstracted, while areas of interest can be expanded and investigated.

Because they are derived from and linked to the reality of the application code, architectural models become real. Models are no longer abstract concepts; they are intrinsically linked to actual source code.

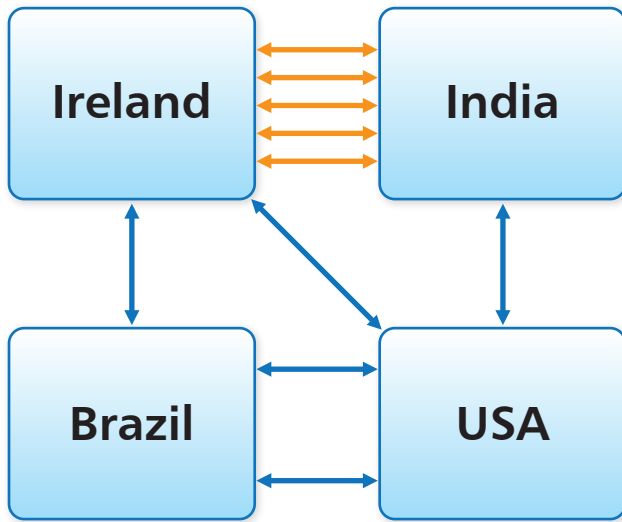
### Example Value 2: Manage Applications

The use of metadata filters help architects to monitor the adherence of the application code to corporate standards and requirements. This can be achieved through APM measurements that are filtered by the metadata concepts the architect has defined. Application cost, complexity, size, business value, risk, complexity, architectural quality, and other measurements and can associated to underlying sources and filtered by contexts that are meaningful.

The linkages of various ontologies are useful for associating measurements from other management paradigms back to the code. For instance, an application performance monitoring technology may identify frequent downtime for an application. The linkage of the configuration item for the down application through metadata to the underlying source code allows architects, operations, and development professionals to trace root causes of issues to the code itself.

### Example 3: Correct Architectural Inefficiencies in the Application Portfolio

When you can view and measure the application portfolio through metadata filters, you are able to spot weaknesses that should be corrected. For instance, you may use metadata about geographic control of the application portfolio to spot dependencies. You may see that code managed in Ireland has a particularly large number of dependencies with code managed in India. This can lead to slower and more complex development as teams must constantly communicate to avoid unintended impacts. This discovery may lead managers to re-architect the application to eliminate these complexities.



Architects can also begin to spot digressions from other architectural standards, like appropriate layering of UI, logic, and data access.

## Conclusions

Most architects are confronted with the reality of large and complex portfolios of existing applications. They must regain control over these systems and, where appropriate, encourage a return to architectural standards that have eroded over time. The first step to moving to more effective architectures for existing applications is to understand where you currently stand. A metadata-driven approach to abstracting applications makes this possible.

### About Micro Focus

Micro Focus, a member of the FTSE 250, provides innovative software that allows companies to dramatically improve the business value of their enterprise applications. Micro Focus Enterprise Application Modernization and Management software enables customers' business applications to respond rapidly to market changes and embrace modern architectures with reduced cost and risk.

For additional information please visit: [www.microfocus.com](http://www.microfocus.com)