

Weg von COBOL, aber nicht zu Microservices: Mercator-Leasing setzt auf einen modularen Java- Monolithen

Mercator-Leasing finanziert als bankenunabhängige Leasinggesellschaft in Deutschland Wirtschaftsgüter von Maschinen und IT bis hin zum Bike-Leasing. Das dafür zentrale, über Jahrzehnte gewachsene Kernsystem Leasco auf Basis von COBOL und Oracle soll schrittweise abgelöst werden. Im Projekt Esprit entsteht dafür gemeinsam mit Gofore eine neue cloudnative Plattform – bewusst als modularer Monolith statt als Microservices-Architektur.

*von Michael Bock, MLF Mercator-Leasing
und Mario Herb, Gofore*

Leasco entkoppeln, ohne den Betrieb zu stoppen

Das Altsystem Leasco (COBOL/Oracle) bündelte sämtliche Domänen – von Geschäftspartner-Stammdaten über Bonität und Vertragsverwaltung bis Finanzbuchhaltung



Michael Bock, MLF Mercator-Leasing

Quelle: MLF Mercator-Leasing

und Refinanzierung – in einem Monolithen. Statt einer risikoreichen Komplettmigration wurde das System entlang fachlicher Domänen schrittweise entkoppelt:

Geschäftspartnerverwaltung: Mangels Altsystem-APIs wurde die GP-Verwaltung zuerst herausgelöst. Die Schreibhoheit liegt heute im neuen Java-Kern, während Daten synchron über eine abgesicherte Schnittstelle in das Altsystem übertragen werden.

Automatisierte Bonitätsprüfung: Externe Auskunftsteilen wurden regelbasiert angebunden. Die Steuerung der Prüfpfade erfolgt regelbasiert, die abschließende Entscheidung bleibt fachlich verantwortet und wird mit Zeitstempel und Entscheider nachvollziehbar historisiert.

Vertragskern: Der produktive Einstieg erfolgte über das datenintensive Benefit-Geschäft (u. a. Bike-Leasing). Parallel werden SAP für die Finanzbuchhaltung sowie BearingPoint Assets & Finance für die Refinanzierung angebunden.

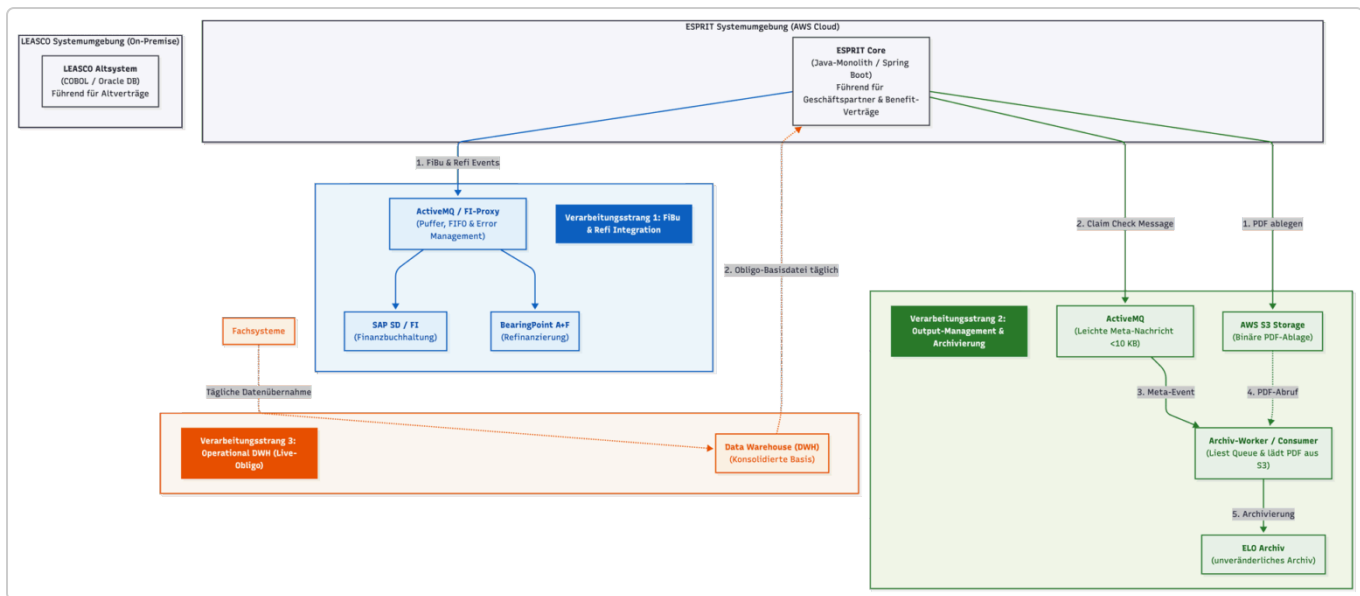
Warum ein modularer Monolith statt Microservices?

Die Wahl von AWS in der Region Frankfurt vereint hohe Skalierbarkeit im regulierten Umfeld mit hoher Agilität: Dank konsequentem Infrastructure as Code via OpenTofu lassen sich neue Stages für parallele Entwicklungs- und Testprozesse auf Knopfdruck automatisiert hochfahren. Das Identity Management erfolgt über einen Managed Identity Provider nach den Standards OIDC und OAuth 2.0. Secrets werden zentral und verschlüsselt in einem Managed Secrets Store verwaltet.



Mario Herb, Gofore

Quelle: Gofore



End-to-End-Zielarchitektur von Esprit mit AWS-Cloud-Kern sowie entkoppelten Verarbeitungssträngen für FiBu, Refinanzierung, GoBD-Archivierung und DWH-Anbindung.

Quelle: MLF Mercator-Leasing GmbH & Co. Finanz-KG und Gofore

Statt verteilter Microservices fiel die Architekturentscheidung bewußt auf einen modularen Monolithen auf Basis von Java, Spring Boot, Hexagonaler Architektur und taktischem Domain-Driven Design. Innerhalb der Modulgrenzen bleiben fachliche Transaktionen damit lokal und ACID-konsistent – dort, wo in einer Microservices-Architektur verteilte Transaktionen oder Saga-Kompensationen nötig wären. Das reduziert die operative Komplexität erheblich und wahrt die fachliche Kohärenz im regulierten Umfeld. Über Systemgrenzen hinweg, etwa zu SAP und BearingPoint A&F, arbeiten wir bewusst mit asynchroner Konsistenz und definierten Abgleichverfahren. Den übergeordneten regulatorischen Rahmen bilden dabei die Mindestanforderungen an das Risikomanagement (MaRisk) mit ihren Vorgaben zu Nachvollziehbarkeit, Kontrolle und Ausfallsicherheit.

Mario Herb, Gofore



Mario Herb ist Principal Software Architect bei Gofore ([Website \(https://gofore.com/en/\)](https://gofore.com/en/)). Er begleitet Unternehmen bei der Modernisierung komplexer IT-Systeme und entwickelt skalierbare Softwarearchitekturen für regulierte Umgebungen. Im Projekt Esprit war er maßgeblich an Architektur und Umsetzung beteiligt.

“ Durch die saubere Isolation über die hexagonale Basis bleibt die Architektur jedoch voll evolutionsfähig: Sollten einzelne Domänen künftig eine unabhängige Skalierbarkeit oder eine stärkere Entkopplung paralleler Entwicklerteams erfordern, lassen sie sich mit überschaubarem Aufwand als dedizierte Microservices herausbrechen.“

Die Anbindung von SAP und BearingPoint erfolgt asynchron über ActiveMQ. Message Groups gewährleisten FIFO-Reihenfolgen, Queues puffern Lastspitzen, Dead Letter Queues isolieren fehlerhafte Nachrichten („Poison Pills“), ohne den Gesamtdurchsatz zu beeinträchtigen.

Hochlastverarbeitung im laufenden Betrieb

Sollstellung mit mehr als 600.000 aktiven Verträgen: Ein vorgeschaltetes Prüfmodul stellt sicher, dass nur fachlich vollständige Vorgänge in die Verbuchung gelangen. Die Übertragung an die Finanzbuchhaltung erfolgt asynchron und wird erst nach erfolgreichem Abschluss des gesamten Verarbeitungslaufs freigegeben.

Operational DWH & Live-Obligo: Um synchrone Systemabfragen im Tagesgeschäft zu vermeiden, wird die Obligo Basis regelmäßig in einem Data Warehouse konsolidiert. Das aktuelle Gesamtrisiko ergibt sich zur Laufzeit aus dieser konsolidierten Basis und den seither neu hinzugekommenen Vorgängen. Für Störfälle bestehen definierte Rückfallverfahren, die die Verfügbarkeit des Tagesgeschäfts sicherstellen.

Archivierung via Claim Check Pattern: Bei hohem Dokumentaufkommen werden die Dateien direkt in einem Objektspeicher abgelegt. Über die Messaging-Infrastruktur werden lediglich schlanke Metadaten und Referenzen auf die abgelegten Dokumente transportiert. Archivierungs-Worker lösen den Claim-Check ein und übergeben die Dokumente an eine unveränderbare Ablage im ELO-Archiv. Zusammen mit der Verfahrensdokumentation adressiert dieses Vorgehen die Anforderungen an Nachvollziehbarkeit, Unveränderbarkeit und ordnungsgemäße Aufbewahrung im Rahmen der GoBD.

Was wir aus der Migration gelernt haben

Wer die Ablösung eines gewachsenen COBOL-Monolithen als rein technisches Projekt betrachtet, unterschätzt leicht die organisatorische Dimension. Es zeigte sich, dass die Komplexität weniger im Code lag als in der Organisation. Mit dem Herauslösen der Bounded Contexts veränderten sich eingespielte Prozesse und Schnittstellen quer durch das Unternehmen. Neben der technischen Architektur mussten daher auch Organisation, Entscheidungswege und Zusammenarbeit mit dem Projekt Schritt halten.

Eine besondere Herausforderung bestand darin, alle Führungsebenen kontinuierlich in die Transformation einzubinden.

“ *Der Zielkonflikt zwischen zusätzlichen fachlichen Anforderungen und verbindlichen Projektterminen erforderte klare Entscheidungswege und eine transparente, übergreifend akzeptierte Priorisierung.* “

Eine weitere Erkenntnis betrifft den Betrieb. Eine cloud-native Zielarchitektur lässt sich nicht dauerhaft auslagern. Parallel zur technischen Transformation mussten wir gezielt internes Know-how und zusätzliche Kompetenzen für den Betrieb und die Weiterentwicklung der neuen Plattform aufbauen.

Schon heute steht eine hochautomatisierte Plattform: Infrastrukturänderungen sind über Infrastructure as Code versioniert und nachvollziehbar, die Dokumentenarchivierung läuft über einen einheitlichen Mechanismus statt über gewachsene Einzelwege, und fachliche Vorgänge werden innerhalb des Monolithen zentral protokolliert. Abgeschlossen ist die Transformation damit aber noch nicht. Die Transformation wird in weiteren Schritten fortgesetzt. Das Zielbild bleibt der schrittweise und vollständige Rückbau des COBOL-Altsystems Leasco.

Michael Bock, MLF Mercator-Leasing



Michael Bock ist IT-Leiter bei der MLF Mercator-Leasing ([Website \(https://www.mercator-leasing.de/\)](https://www.mercator-leasing.de/)). Er verantwortet dort IT-Strategie, Governance und Infrastruktur und treibt die Transformation der Kernsysteme hin zu modernen Cloud- und Anwendungsstrukturen im regulierten Umfeld voran. Im Projekt ESPRIT leitet er die Ablösung des COBOL-Altsystems Leasco.

Michael Bock, MLF Mercator-Leasing und Mario Herb, Gofore ■

Sie finden diesen Artikel im Internet auf der Website:

<https://itfm.link/250991>



