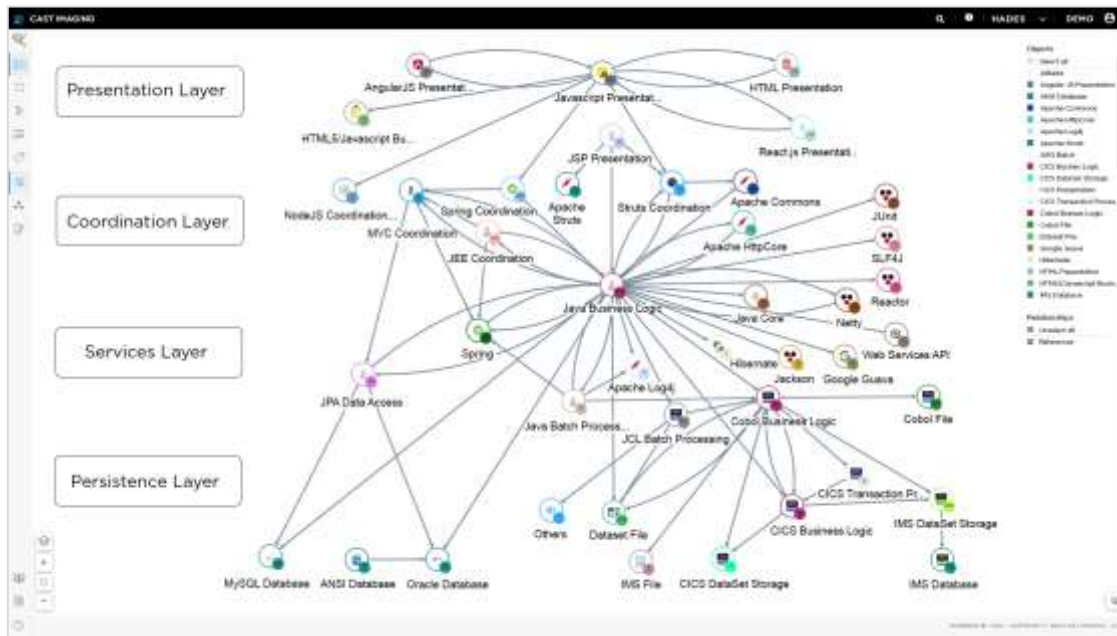


CAST Imaging Vs. vFunction

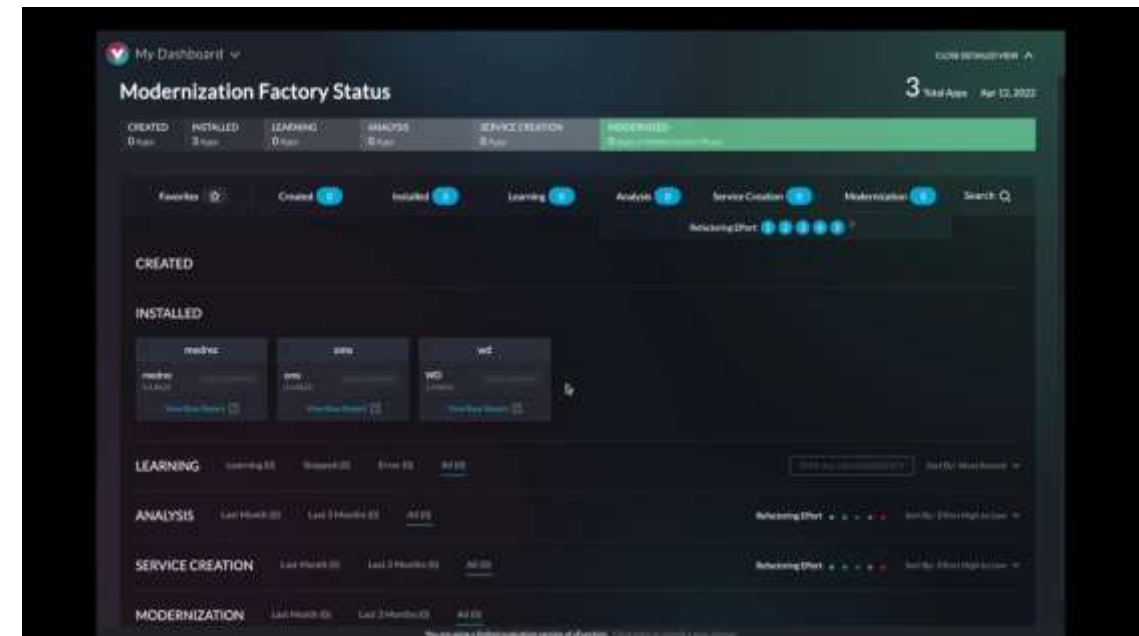
Product description

CAST Imaging helps application development teams to understand, evolve, and modernize applications. It automatically reverse-engineers all database structures, code components, and interdependencies in custom-built applications, down to the tiniest details. It provides interactive and accurate architecture blueprints, data-call graphs, and end-to-end transaction flows in a lightweight web UI with the ability to export details externally.

vFunction is an assessment and modernization tool for Java applications. It assesses Java apps to calculate their modernization capacity using proprietary metrics (based on technical debt, complexity, risk) and then automatically extracts microservices from classes of Java monolith applications. vFunction does not offer any blueprint or visual maps of the application internals.



CAST Imaging



vFunction

CAST Imaging Vs. vFunction

Key functionalities comparison

Functionality	CAST Imaging	vFunction
Technologies, installation, integration		
Technology Coverage	Mobile: Swift, Objective C, Android	Core Logic: Java, .NET (beta as Nov. 2022)
	Web & Frameworks: HTML5, ASP, ASP.NET, JavaScript, Typescript, NodeJS, AngularJS, Vue.js, React.js, Apache, Google, Amazon, Spring, XHTML.	
	Core Logic: C#, .NET, VB, Java, C/C++, SAP/ABAP, Fortran, Powerbuilder.	
	Cloud: AWS, Azure, GCP technologies.	
	Mainframe technologies: zOS Cobol, Micro Focus Cobol, GCOS Cobol, Cobol VMS, CICS, JCL, IMS, RPG, PL/I, Natural (CAST Community), IDMS (CAST Community).	
	COTS: ABAP	
	Database - RDBMS : SQL Server, Oracle, DB2, MariaDB, SQLite, Azure SQL, Informix, MySQL, PostgreSQL, Sybase, Teradata, Adabas (CAST Community). Database - NoSQL: MongoDB, Azure Cosmos DB, AWS Dynamo DB, Elasticsearch, Marklogic, and more	
Light Web UI	Yes	Yes
Agentless	Yes	No (<u>recommended installing in production</u>)
Free of building the application, binary data, and test runs	Yes	No
Application Discovery		
Analysis of intra-technology links	Yes (<u>All Technologies</u>)	Yes (only Java)
Analysis of inter-technology links (App-to-App dependencies)	Yes (<u>All Technologies</u>)	No
Automatic e2e Transaction View (UI→MiddleTier→Database) and Data Call Graphs	Yes	No
Standardized metrics on all objects (cyclomatic/essential complexity, Halstead metrics...)	Yes	No (proprietary metrics)
Application Changes & Modernization		
Micro-services representation	Yes	No
Identification of modernization candidates (refactoring, de-coupling, ...)	Yes (thru interpretation of visuals)	Yes (based on proprietary metrics Debt, Risk, Complexity on <u>classes only</u>)
Automatic creation of microservices with code generation	No	Yes
Model architecture changes	Yes (<u>All Technologies</u>)	No
Root-cause & impact analysis	Yes (<u>All Technologies</u>)	No
Team collaboration and Living Knowledge base		