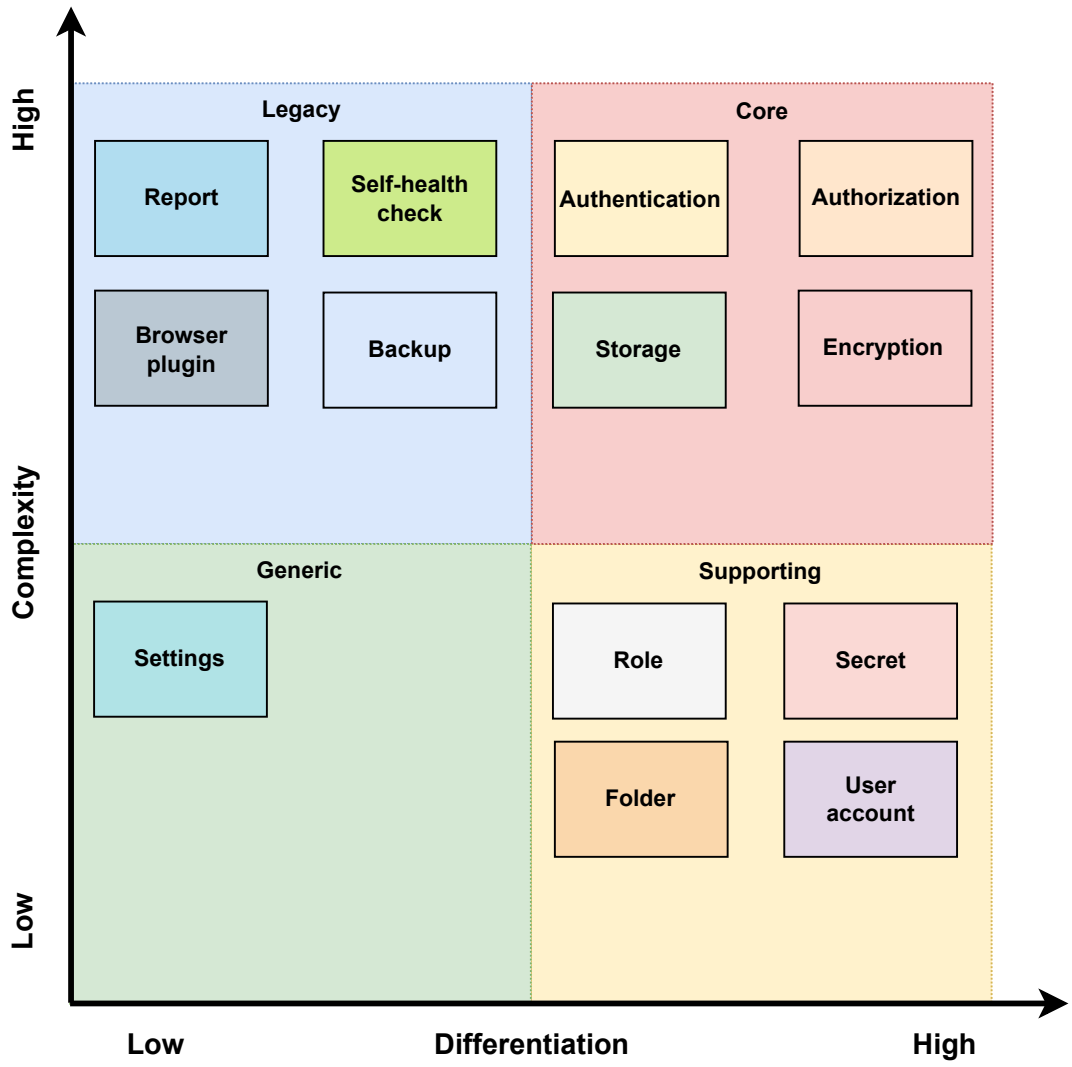
Event Storming

Encryption	Storage	Backup	User account	Authentication	Authorization	Settings	Report	Browser plugin	Role	Folder	Secret	Self-health check
All stored data must be encrypted The connection between frontend and backend must be encrypted The connection between the user and the frontend must be encrypted	The storage must be a database with the ability to encrypt data The storage should be a database with the ability to backup data Convenient connection	The system must be in high availability mode It should be possible to automate backup and upload it to a third-party server	The user must have full control over their account Login and logout user Possibility to change password View user-owned folders and secrets	Multi-factor authentication required Support for SAML, Open ID and LDAP protocols is required Automatic logout when inactive	OAuth 2.0 protocol required Role-based access support Strictly limiting visibility to only the data that the user has access to	The ability for the user to personalize their personal account Ability for the user to set privacy settings Ability for the user to set secret generation settings	Generating reports on leaked secrets Analysis of user access in order to limit unnecessary rights	The plugin must support all current browsers The plugin should be able to intercept the web application authentication form and insert credentials	Multiple roles should exist to separate the access rights of different users Only administrators should be able to manage user roles	It should be possible to create folders that allow users to structure their data The owner of the folder must be able to grant read or edit access to another user	The owner of the folder must be able to grant read or edit access to another user The user must be able to create and generate automatic secrets	The functionality must cover a complete check of all application components Health check should generate error reports

Core Domain Chart



Relations / Mappings