#	Subject	Requirement	Compliance	Details
	Software catalog			
C1	Catalog library	The IDP shall support Github SaaS / Gitlab (SaaS and self hosted) / Azure Devops / Bitbucket Cloud / Bitbucket server		
C2	Catalog library	It is possible to ingest yaml and READMEs in the software catalog		
C3	Catalog library	The IDP shall support ingesting data from common CI tools to track pipeline workflows		
C4	Catalog library	Support for Cloud vendors: AWS, GCP, Azure Ingest data from EKS, Resource Groups, VPC, Virtual Machines, Functions/Lambdas, Buckets, IAM policies		
C5	Catalog library	Ingest catalog data from Argo CD		
C6	Catalog library	Ingest catalog data from Kubernetes		
C7	Catalog library	Ingest data from Jira and all the tickets, status, projects		
C8	Catalog library	Ingest data from Pagerduty, Opsgenie		
C9	Catalog library	Ingest tickets from ServiceNow		
C10	Catalog library	Ingest CVE related to Git projects from Snyk, Sonarqube, Wiz		
C11	Catalog library	Ingest data from Cost tools: AWS Cost Explorer, Kubecost, OpenCost		
C12	Catalog library	Ingest data from Terraform Cloud to represent tf states and projects		
C13	Ingestion Method	Ability to ingest data from any webhook from any 3rd party solution		link to doc
C14	Ingestion method	Ability to ingest data from API		link to doc
C15	Ingestion method	Ability to ingest data from Gitops with a yaml file in a repo		
C16	Ingestion method	Ability to ingest data from Terraform: ingest catalog entities from Terraform ingestion		
C17	Ingestion method	Ability to create or edit a catalog item directly in the UI		
C18	Extensibility	Ability to add a new plugin or integration by ourselves from an unsupported vendor or data source.		provide link to documentation
C19	Extensibility	Ability to add custom properties to an existing data model from a vendor. For instance ability to add the subscription description or domain or user to any object in the catalog		
C20	Extensibility	Ability to map a data source to a new custom property using JQ. For instance add a new metadata that is not present in the IDP by default for the 3rd party vendor's API		
C21	Extensibility	Extending an integration to additional properties shall not require front end or backend coding		
C22	Extensibility	Ability to create a new plugin or integration by ourselves		
C23	Data model	It is possible to display properties from various entities of various types in the catalog in a single view *For instance: display the CPU usage of the K8S deployment of a service ingested in Github* *Or how many vulnerabilities have been found by a security scanner in the service view without leaving the page*		
C24	Data model	Ability to create relations between any elements of the catalog		describe how flexible can the mapping be and what prerequestes are required
C25	Data model	Ability to create relations between any elements of the catalog based on similarities in their name		

#	Subject	Requirement	Compliance	Details
C26	Data model	Ability to create relations between any elements of the catalog based on similarities in any of their properties (tags, CRD or any other custom property)		
C27	Data Model	Ability to ingest data from various sources into a single entity. For instance Ingest a service from Github and enrich a few of its properties using Gitops or API ingestion		
C28	Data Model	A change in the data source is reflected in real time in the catalog and should be event-based (not through scheduled syncs) Example: A change in a Kubernetes environment or creation of a new EKS cluster should be reflected within seconds in the catalog		
C29	Data Model	The catalog shall offer a graph visualization of the intereralted entities in the catalog. *Such as service upstream/downstream, or service under a cloud deployment*		
C30	Data Model	It is possible to display calculated properties in the catalog view from multiple other properties of the entity		
C31	RBAC	We can define granular controls on who can see what pages of the catalog		prodive doc
C32	RBAC	We can define locked filters or view so a user can only see certain entities in a catalog, while another user can see others based on his role or team he belongs to		
C33	RBAC	The catalog supports dynamic filters, showing the user catalog entities that are specific to him (such as the user's Pull Requests, the user's services)		
C34	RBAC	Certain views in the catalog can be locked to prevent changing filters or presenting hidden data		
	Scorecards			
S1	Scorecards	The IDP shall allow to create flexible scorecards from any data source		
S2	Scorecards	Data from multiple data source can be part of a single Scorecard *For instance: a service scorecard shall ensure there is on call engineer in Pagerduty, a README in Github and less than 5 high vulnerabilities from Snyk*		
S3	Scorecards	The IDP shall offer consolidated views of Scorecards with Group By options, such as "group by team", "group by Domain" or any other property		
S4	Scorecards	It is possible for a team to see the aggregated statistics of all rules in a summarized manner		
S5	Scorecards	The IDP shall be able to automatically create Jiras for Scorecards that are not met		
S6	Scorecards	Scorecards criteria can include the following operators to interrogate any property: =, <>, >, contains, endWith, isEmpty, beginsWith		
S7	APIs	Scorecards can be queried from APIs and integrated into a CI. We can block a CI build if a service is not meeting a certain scorecard level		
S8	Notifications	The IDP shall be able to send notifications in case of a degradation of a scorecard		
S9	Notifications	The IDP shall be able to automatically create Slack notifications providing summaries of Scorecards that are not met		

#	Subject	Requirement	Compliance	Details
	Developer self-service actions			
A1	Self Service Actions	The IDP shall offer Self Service actions capabilities		
A2	Advanced Forms	The Self Service action shall offer to create any custom form for developers, that shall include: free text with Regex control, number, boolean toggle, email, Data & Time, encrypted secret, URl, dropdown and others input types		
A3	Advanced Forms	The Self Service form shall include selection of entities in the catalog		
A4	Advanced Forms	The forms shall offer the ability to filter the dropdown's available options based on a previous input		
A5	Advanced Forms	The form shall provide you a choice of entities that is filtered based on previous answers in a flexible manner. *To illustrate this, if a user picked AWS Region US east 1, the form that only offer the available EKS clusters within that region dynamically available in the catalog*		
A6	Advanced Forms	The forms offer the option to hide/show a form based on a previous use input		
A7	Advanced Forms	The form shall offer to filter the dropdown's available options based on the properties of the entity on which the action is performed		
A8	RBAC	The form shall offer to filter the dropdown's available options based on properties of the user that executes the action		
A9	RBAC	It shall be possible to configure who can trigger a self service action based on the team they belong to		
A10	RBAC	It is possible to configure an approval in a self service action		
A11	RBAC	It is possible to dynamically ensure that the user who is running the action is assigning the Microservice to his team in the form. He should not be offered to fill someone else's team		
A12	Backend	It is possible to trigger a CI pipeline job directly		specify which CIs is supported
A13	Backend	It is possible to trigger a generic Webhook		
A14	Backend	It is possible to trigger a CI that is on premise, not accessible from a direct connection, using an agent		describe how this is supported and if there are limitations in the scenarios supporters or prerequisites on the customer's side
A15	Backend	It is possible to chain Self Service actions		
A16	Backend	Ability to edit a yaml file within a github repo through a form in the IDP		
A17	Dev Experience	The developer can track the progress of a self service action, such as: pending approval, in progress, failure or success		
A18	Dev Experience	It is possible to add custom messages for the developer to track the progress of self service actions		
A19	Trigger Automation	It is possible to configure an automatic trigger of a self service action based on a entity change or a scorecard degradation		
	Dashboards, Pages & Customization			
D1	Customization	The IDP shall allow the creation of custom pages or dashboards		
D2	Customization	it is possible to control the Entities properties that are displayed in the UI		
D3	Customization	We can create multiple views (more detailed/less detailed) for the users, showing or hiding properties for different types of users		

#	Subject	Requirement	Compliance	Details
D4	Customization	Custom Dashboards can have a custom format, such as: creation of html text, iFrame or widgets		
D5	Customization	The structure of the custom dashboard is configurable and flexible		
D6	Customization	The custom pages can represent custom dynamic data from the catalog		
D7	Customization	The order of the pages can be modified		
D8	Customization	Pages of the catalog can be grouped or structured		
D9	Customization	Customizing a page view does not require front end coding		
D10	Customization	It is possible to display data any data source of the catalog in a single Dashboard page		
D11	RBAC	The permissions of who can seen a custom page are configurable (per team, per user)		
D12	RBAC	The views of a custom page can be dynamically configured based on who is the logged in user ("my Pull Requests", "my services")		
D13	RBAC	The IDP shall offer the ability to provide different views and experiences for different user roles: Managers, Developers, Product Managers		
	General Requirements, Networking & Security			
G1	General Requirements	The solution shall support Okta, Azure AD, Google Workspaces		
G2	General Requirements	The IDP shall support multiple tenants for multiple independant Business Units		
G3	General Requirements	It shall be possible to store the IDP's configuration as a code in a Git repo		
G4	General Requirements	It shall be possible to store the Self Service actions and Scorecards as a Code in a Git repo		
G5	General Requirements	It shall be possible to deploy the IDP using IaC such as Terraform or Pulumi		
G6	API	The IDP shall have extensive APIs to change the IDPs configuration		Send documentation
G7	API	The IDP shall have extensive APIs to ingest new entities in the catalog		
G8	API	The IDP shall have extensive APIs to update entities properties in the catalog		
G9	API	It is possible to trigger a self service action from API, and integrate it in a CI for instance		
G10	Search	The IDP shall offer an effective Global Search to find any entity in the entire catalog based on its name only		
G11	Search	The IDP shall offer advanced search capabilities over APIs, combining multiple criteria (OR, AND, CONTAINS) as well as filtering		
G12	Security	It is possible to deploy the catalog without any ingress connectivity to our network, by Push data only		
G13	Security	The solution does not require to host secrets of our systems in the vendors' cloud		
G14	Security	The vendor shall be SOC 2 compliant		send doc
G15	Security	The vendor shall be ISO 27001		send doc
G16	Documentation	The documentation shall be comprehensive and publicly available		Share link

#	Subject	Requirement	Compliance	Details
	Services			
E1	Customer success	The vendor shall offer postsales services to help us implement the IDP		
E2	Customer success	The vendor shall offer Slack support		
E3	Support	All offered plugins and integrations mentionned in this RFP answer shall be actively supported by the vendor		