Color codes: *Fulfilled requirement* | *Partial fulfillment* | *Uncertain fulfilment* | *No fulfillment*

	ProM	RapidProM	RuM	Cortado	Smyrida	Disco	Apromore	SAP Signavio	Celonis	PMAKNIME	DAPM prototype
Confidentiality/data integrity	As a desktop application that does not need online connection DURING the analysis, it can run on a restricted environment. In addition, the code can be inspected.	As a desktop application that does not need online connection DURING the analysis, it can run on a restricted environment. In addition, the code can be inspected.	As a desktop application that does not need online connection DURING the analysis, it can run on a restricted environment. In addition, the code can be inspected.	As a desktop application that does not need online connection DURING the analysis, it can run on a restricted environment. In addition, the code can be inspected.	In principle the system can be deployed anywhere, but would require developer/sysadmin skills to achieve this.	As a desktop application that does not need online connection DURING the analysis, it can run on a restricted environment.	On premis might be an option? Also, different locations?	On premis might be an option? Also, different locations?	On premis might be an option? Also, different locations?	The desktop application runs locally so no data sharing. If deploying for remote execution, then data is transfereed to KNIME's server. Deployment into existing clusters is supported, but requires developer/sysadmin skills.	For miner nodes, confidentiality is ensured via shadowing. For repository, the current prototype does not have access control.
IP protection of algorithms	Possibility to include packages with different licenses including closed sources.	Possibility to include packages with different licenses including closed sources.				The application is closed source.	While open-source version exists, the deployed version has closed-source extensions.	The application is closed source.	The application is closed source.	Private spaces are available.	Microservices allow to maintain algorithms at provider's end. Also shadowing can be done on executables/binaries.
Interoperability	Supports some degree of interoperability via standard exchange format (eg XES, PNML, BPMN) but with manual data transfer and manipulation needed.	Supports some degree of interoperability via standard exchange format (eg XES, PNML, BPMN) but with manual data transfer and manipulation needed.	Supports some degree of interoperability via standard exchange format (eg XES) but with manual data transfer and manipulation needed.	Supports some degree of interoperability via standard exchange format (eg XES) but with manual data transfer and manipulation needed.	Probably achievable due to architecture, but not documented.	As long as "AirLift" protocol is matched (https://fluxicon.com/blog/2014/07/disco-1-7-0/), very limited integrations.	Specific integrations have been done (see https://apromore.com/newsroom/apromore-announces-seamless-integration-with-mulesoft-rpa-manager/). Maybe other integrations can be requested?	"Multiple integration capabilities" mentioned (https://www.signavio.com/products/process-intelligence/). Also "Ingestion API"s.	Via Extractor Builder (https://docs.celonis.com/en/data-integration.html).	Supports some degree of interoperability via standard exchange format (eg XES, PNML, BPMN) but with manual data transfer and manipulation needed.	Via REST API for creating and wiring miners.
Extensibility	Plug-in architecture, extra packages can be added.	RapidMiner (platform behind RapidProM) has a plug-in based architecture.	Open source, can be extended (requires developer).	Open source, can be extended (requires developer).	In principle, based on REST API, but no standardized calls.					New extensions can be installed.	New miner nodes can be added with few lines of source code.
Scalability/Elasticity	Vertical scaling (more computational power), in principle possible but no elasticity (increasing/reducing the power based on needs).	Vertical scaling (more computational power), in principle possible but no elasticity (increasing/reducing the power based on needs).	Vertical scaling (more computational power), in principle possible but no elasticity (increasing/reducing the power based on needs).	Vertical scaling (more computational power), in principle possible but no elasticity (increasing/reducing the power based on needs).	Horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.	Vertical scaling (more computational power), in principle possible but no elasticity (increasing/reducing the power based on needs).	We assume horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.	We assume horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.	We assume horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.	Only for remote deployment: we assume horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.	Horizontal scaling possible (load of users spread on many instances). Vertical scaling and elasticity possible in the cloud.
Configurability/Flexibility		The process panel allows to design the dataflow to be processed without any programming.								The workflow editor allows to design the dataflow to be processed without any programming.	Current version of the frontend allows wiring but limited to one miner at the time.
Reproducibility/Provenance		The process panel allows to see how a certain result has been produced, but history is lost if the dataflow is new.			"Recipies" allow reproducibility of filters (https://fluxicon.com/blog/2014/07/disco-1-7-0/).				Filters can be seen and allow some reproducibility/provenance to be exposed.	The workflow editor allows to see how a certain result has been produced, but history is lost if the dataflow is new. Intermediate results can be inspected.	Reprodubility possible as long as services are exposed, provenance data stored in metadata files.
Reusability		Workflows can be exported and reused, granted that the packages are available.			"Recipies" can be reused (https://fluxicon.com/blog/2014/07/disco-1-7-0/).					Workflows can be exported and reused, granted that the extension is installed and available.	The backend does cater for reusability but the frontend does not allow reusability.