Concern (Identifier: Description)		*Con#2: Which architectural pattern should the system use?*
Ranking criteria (Identifier: Name)		*Cr#1: scalability* *Cr#2: decoupling*
Options	***Identifier: Name***	*Con#2-Opt#2: Publish/Subscribe*
	Description	*PubSub architecture where the publishers are the external databases (fab_data & raw_data), the topics are related to event's messages architecture and subscribers are the end users.*
	Status	*This option is decided.*
	Relationship(s)	-
	Evaluation	*Cr#1: Pub/Sub architecture provides scalability, if one topic is under strain we can add other computational power to it.* *Cr#2: Pub/Sub architecture provides de-coupling.*
	Rationale of decision	*This option is decided because it satisfy our requirements and solves the criteria in optimal manner.*
	Identifier: Name	*Con#2-Opt#2: Client/Server*
	Description	*Communication in this architectue among the components is based on client/server pattern.*
	Status	*This option is rejected.*
	Relationship(s)	-
	Evaluation	*Cr#1: In order to achieve scalability we need to set up load balancing tecniques.* *Cr#2: Clients are stricty coupled with server's architecture.*
	Rationale of decision	*This option is rejected because it doesn't satisfy the requirements.*