

nEngineBroker SimpleAllowArbiter SimpleDenyArbiter reprocessor anager ence Manager Preprocessor AllowArbiter Den yArbi ter «abstract» ation Table Arbiter + mergeDecisions(Response[0..*]) : Response DecisionEngineBroker getMechanism(string): Mechanism instantiateDecisionEngine(Mechanism):Status listMechanisms():string[0..*] notifyEvent(Event) :Response «use» removeDecisionEngine(string):Status «use» **PDPManager** PersistenceManager + evaluate Predicate (String): boole an + loadPDPState():void getContainerForData(Data) :Container[0..* persistPDPState():void getDataInContainer(Container):Data[0..*] processPersistenceEvent():void «interface» **IPDP** oyMechanism (Mechanism):Status PolicyPreprocessor rtMechanism(string):Mechanism yEvent(Event) :Response Ev entPreprocess or checkMechanismSanity(Mechanism):Status ceMechanism (string): Status processMechanism (Mechanism): Mechanism processEvent(Event) :Event + revokeMechanism(string):Status «persistent» TranslationTable DataEventMap :DataEventMap StateEventMap :StateEventMap get():void iterate():void + put() :void + remove():void





