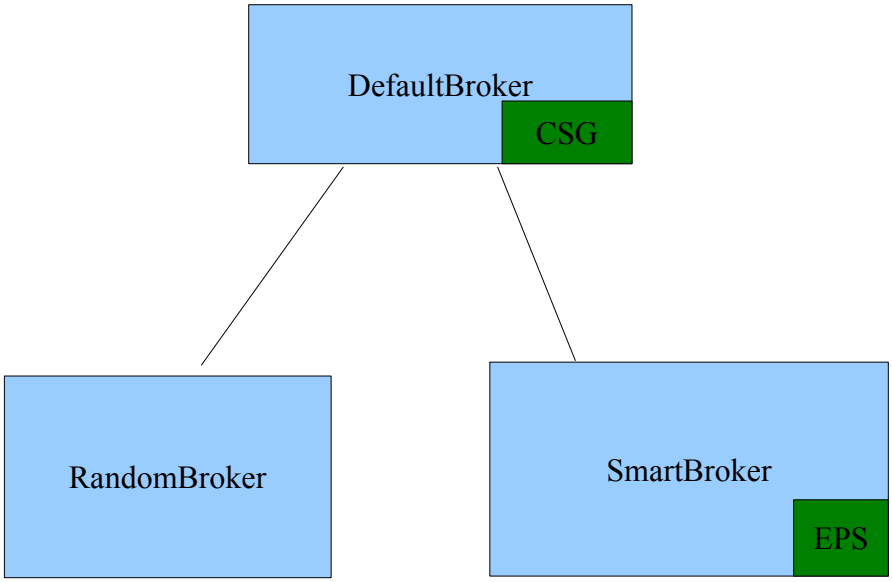

http://www.ogf.org/documents/GFD.80.pdf

- Execution Planning Services (EPS)An Execution Planning Service (EPS) is a service that builds “schedules,” where a schedule is amapping (relation) between services and resources, possibly with time constraints. A schedulecan be extended with a list of alternative “schedule deltas” that basically say “if this part of theschedule fails, try this one instead.”An EPS will typically attempt to optimize some objective function such as execution time, cost, reliability, etc. An EPS will not enact the schedule; it will simply generate it. The enactment of aschedule is typically done by the JM. An EPS will likely use information services and CandidateSet Generators (CSG, see below). For example, first call a CSG to get a set of resources, then getmore current information on those resources from an information service, then execute theoptimization function to build the schedule.

- 3.4.6.2 Candidate Set Generator (CSG)The basic idea is quite simple: determine the set of resources on which a unit of work canexecute—“where is it possible to execute?”, rather than “where will it execute?” This mayinvolve issues such as what binaries are available, special application requirements (e.g., 4GBmemory and 40GB temporary disk space, xyz library installed), and security and trust issues (“Iwon’t let my job run on a resource unless it is certified Grade A+ by the Pure ComputingAssociation,” or “they won’t let me run there until my binary is certified safe,” or “will theyaccept my credit card?”).A Candidate Set Generator (CSG) generates a set of EPRs of containers in which it is possible torun a job described by a JSDL document. The set of container resources to search over may eitherbe a default for the particular service or be passed in as a parameter.We expect CSGs to be primarily called by EPSs, or by other services such as JMs that areperforming EPS-like functions. We expect CSGs to use information services, to access jobs toacquire appropriate pieces of the job document and to interact with provisioning and containerservices to determine if it is possible to configure a container for a particular execution.