System Availability

1. System Names
   1. Environment
      1. TEST
      2. DEVL
      3. DEMO
      4. ~~LOAD~~
      5. ~~PROD~~
   2. System
      1. PAS
      2. CRM
      3. Client
      4. Billing
      5. Actuarial
      6. Ratabase
      7. MarketingAutomation
      8. DocMgmnt
      9. UserMgmnt
      10. Claims
      11. BI
      12. Reporting
      13. FrontEnd
      14. ESP
      15. BackOffice
      16. Accounting – add to stysem
   3. Labels
      1. AdobeDBServer
      2. AdobeServer
      3. AppServer
      4. AuthServer
      5. AuthDBServer
      6. BatchServer
      7. BillingServer
      8. BuildServer
      9. CICSEnv
      10. CassandraDB
      11. ChefServer
      12. CloudServer
      13. DB2Env
      14. DBMonitor
      15. DBServer
      16. DevOpsServer
      17. DiscServer
      18. DocServer
      19. DocMgmntServer
      20. DocRepoServer
      21. Domain
      22. ElasticDB
      23. Environment
      24. ExternalService
      25. Feature
      26. FrontEndServer
      27. GatewayServer
      28. KnifeServer
      29. LexisNexisService
      30. LPAR
      31. Mainframe
      32. MajescoServer
      33. MongoDB
      34. PASServer
      35. Program
      36. QServer
      37. Queue
      38. RatabaseServer
      39. RESTEndPoint
      40. RESTRequest
      41. RepoServer
      42. Scheduler
      43. SmartyStreetsService
      44. SQLServer
      45. TimeServer
      46. VM
      47. VMCluster
      48. VagrantServer
      49. Hardware
2. Notes
   1. How to manage Integration and Demo?
      1. Add a setup step for Int (after each system is setup)
   2. Add REST endpoints to Features
   3. How to manage different systems may connect to different environments?
3. Feature
   1. Purchase
      1. What parts of the system are required for purchase?
      2. How to they connect?
4. To Do
   1. Add read of RTG tables to PAS
   2. Add Ratabase middle layer
   3. Graph Gist
   4. Billing
   5. Add Accounting system
   6. Frontend
      1. CMS
      2. DB
   7. PAS Logging
   8. Update Constraints
   9. Doc Management
      1. App server
      2. Repo server
      3. Xnet (MF)
   10. Database monitors
       1. Cassandra monitor is on-site and supports all environments
5. Cool Stuff
   1. Connections to other applications
      1. Queues
         1. Relationship can show RabbitMQ queue binding key. This would show exactly what is being sent to a given system
         2. C:\Users\tsojmw\test>curl -i -u admin:shelter1 http://dcdemoqueuesrv1:15672/api/
         3. queues/%2f/crm/bindings > bindings.txt
      2. REST endpoints
         1. http://dctestappsrv1:8083/api/resource.json
   2. Info on VMs
      1. Call to VMware to get live info on VM (getallvms.py)
   3. Info on Bridge Program
      1. Read config file from either:
         1. Chef
         2. RabbitMQ server
6. Model system
   1. Connects to modeling
   2. Show development over time
      1. Show where each connection is made from development environments
   3. Business function
      1. Quote
      2. Purchase
      3. Policy inquiry
   4. Software
      1. Integration points
         1. Queues
         2. REST endpoints
   5. Domains
      1. Policy Admin
   6. Schedule (ESP)
      1. PAS batch
      2. Claims batch
   7. System infrastructure
      1. System
         1. Say
      2. Platforms
         1. VMware
         2. Mainframe
         3. Cloud
      3. Servers
      4. Middle ware
         1. DB/2
         2. WebSphereMQ
         3. RabbitMQ
         4. Etc.
      5. Status sources
         1. File
         2. Chef
         3. Nagios
7. Modeling Thoughts
   1. CONNECTS\_TO – used to show connections
   2. DEPENDS\_ON – required for functioning
   3. RUNS\_ON – shows dependencies
8. Cypher dump entire database
   1. Must be root
   2. usr@term: bin/neo4j-shell -c dump > ./backup/$(date +"%y%m%d\_%H%M%S").cypher