Enhance our Geo spatial demo in following ways

1) Modify feeder class to publish the data to mainframe in addition to kafka

* find feeder class
* how to publish to mainframe==db?

2) Integrate Delta server to consume the data from mainframe to grid in addition to kafka

* Verizon <https://github.com/GigaSpaces-POCs/VerizonMainFrame>
* extract part of code to demo data load + setup db ???

3) Demo web ui with the spatial and real time query capability

* <https://docs.gigaspaces.com/xap/12.3/dev-java/indexing-geospatial.html>
* <https://docs.gigaspaces.com/xap/12.3/dev-java/query-geospatial.html>
* create pu to insert objects with geo data into space and query via 2 types of queries – spatial (using geo data) and direct ??

4) Integrate AnalyticsXtreme plugin to push data from grid to HDFS/Hive

<https://docs.google.com/document/d/1YxWw3GMOc7N3bMzjV-vg7iUz9UxCsrCO7LLOmd9k6kE/edit?usp=sharing>

5) Demo Zeppelin to run query on both Speed layer and Batch layer(HDFS/Hive)

<https://docs.google.com/document/d/1YxWw3GMOc7N3bMzjV-vg7iUz9UxCsrCO7LLOmd9k6kE/edit?usp=sharing>

6) Demo Spark job on the data and produce some out put to messaging system(Kafka/MQ) <<== maybe we should see if we can have streaming demo

submit job - *./insightedge-submit --class com.gigaspaces.SparkDemo --master spark://localhost:7077 ../../../Crowley-status/standalonesparkjob/target/standalone-spark-job.jar*

start kafka - *kafka-server-start /usr/local/etc/kafka/server.properties*

get consumer console - *kafka-console-consumer --bootstrap-server localhost:9092 --topic test --from-beginning*

filter lines containing ‘spark’ from file ‘testfile.txt’

7) Demo Standalone Spark job running against IE

8) Bonus Demo Tableau integration