**Reviewed by:** Group G06 (*Group Members: Rooma Rathore, Rohini Prinja*)

**Paper Reviewed:** Scalable Trigger Processing  
**Narrative and Presentation by:** Group G04 (*Group Members: Betsy George, Vijay Gandhi*)

**Slides Review Points:**
- Objective is essentially to come up with scalable trigger system, this not only includes minimizing time taken to process trigger but many other things like: how triggers should be stored, indexed etc

- Another motivation is also lack of trigger systems that scale well in all the modern DBMS.

- Related work may also talk about Marking based predicate indexing scheme, RPL system, DIPS System, Datex and possible comparison between these systems

- Validations: The architecture described is implemented as an Informix Datablade. Authors say that it is significant advance over what is available. So this can be validation of the work done.

- Rewrite today can also involve: Stats on how much triggers are used today, if yes to what extent is this kind of an architecture implemented in current DBMS if at all.

- In the Trigger-Man Architecture slide, there should probably be text explaining the main components and the diagram can then be used to explain the flow of information in those components.

- The slides do not talk about How to store expression equivalent classes? I am not sure if you are going to cover it orally when explaining the slides on signatures.

- Are there any assumptions in the paper that you do not agree with?

**Narrative Review Points:**
- Can you give an example of such an internet application where users will create triggers (in the first paragraph)?

- There can be a small section in the narrative explaining the related work in this area and how this architecture solves some of the problems in previous systems. In conjunction with this, it can also cite future directions of the work.