[1, N) = wenn z.B. N = 10 und Entitäten 1, 2, …, 10 heisst das, dass mindestens 1 und maximal 9 Entitäten z.B. BC nicht erhalten, bzw. dass wenigstens 1 (bis maximal 9, aber nicht alle (das wären ja 10)) den BC erhalten haben (es existiert eine Entität, die den BC nicht erhalten hat).

Broadcasting

1. BC\_NEW\_JOB
   1. **Normalfall**: alle erhalten BC (Ɐ(E): E.receive(BCNewJob)
      * Job wird auf Liste getan
      * Jobliste wird sortiert
      * Job wird zu gegebener Zeit ausgeführt
   2. **Ausnahme 1**: [1, N) erhalten BC nicht (Ǝ(E):NOT(E.receive(BC)))
      * Wenn Jobliste leer: wird erst auf nächsten BC reagieren. Sonst wird womöglich anderer Job ausgeführt auf dem Node, der BC nicht erhalten hat
      * **Wenn BC reinkommt für gleichen Job**: alles ok!
      * **Wenn BC reinkommt für anderen Job**: **Konflikt!** 
        1. Einfach lösbar: job1.compareTo(job2) 🡪 der Job, der vor dem anderen kommt, wird zuerst ausgeführt 🡪 d.h. der Executor, der den späteren Job ausführt, bricht diesen ab und räumt Daten auf.
   3. **Ausnahme 2**: niemand erhält BC ([1, N] erhalten BC nicht, Ɐ(E):NOT(E.receive(BC)))
      * Submitter wartet timeToLive Zeiteinheiten, bis BC\_NEW\_JOB nochmals gesendet wird.
      * Submitter wiederholt BC\_NEW\_JOB N Mal