**What is the problem?**

In 9 April 2001, 911 emergency call system outages that affected all of Washington State and parts of Oregon just before midnight because of a “technical error by a third-party vendor” called by telecom provider CenturyLink for these states.

The company estimated that approximately 4,500 emergency calls to 911 call centers went unanswered during the course of the Washington State outage. During the two-hour Oregon outage, which affected some 16,000 phone customers. Also, emergency calls being made in California, Florida, Minnesota, North Carolina, Pennsylvania and South Carolina, in total, some 6,600 emergencies call over a course of six hours.

**What causes this problem?**

The FCC says the cause of the outage was a preventable “software coding error” in a 911 Emergency Call Management Center (ECMC) automated system in Englewood, Colorado, operated by Intrado, a subsidiary of West Corporation.

Ordinarily, Intrado's automated system assigns a unique identifying code to each incoming call before passing it on—a method of keeping track of phone calls as they move through the system. But on April 9, the software responsible for assigning the codes maxed out at a pre-set limit; the counter literally stopped counting at 40 million calls. As a result, the routing system stopped accepting new calls, leading to a bottleneck and a series of cascading failures elsewhere in the 911 infrastructure.

The system monitoring software indicated that the problem was a “low level” problem and it also prevented an automatic rerouting of 911 emergency calls to Intrado’s ECMC facility in Miami; it did not immediately alert anyone that emergency calls were no longer being processed.

**How CAD avoid this problem?**

If a server becomes unavailable for some reason during normal operations, its functions are taken over by another server which is capable of continuing its work. In some cases, the second server will need to have details of the current state of interactions with the users of the first server so that it is able to continue dealing with these users seamlessly.

Rahul looked at several successful systems in larger cities that had coped well as the population increased. He learnt that the growth of these systems depended on them being built on a scalable and extensible architecture right from the beginning of the project. They also used common standards and protocols to make them both *future-proof* and compatible with other information systems. Successful examples included *commercial software*, cloud-based SaaS, and *custom software* developments, but all were built on a *scalable architecture* with the nal choice of product/service/development depending on the requirements of each city.