Problem: Modern systems increasingly consume probabilistic estimates. Programming with estimates is difficult and error-prone.

Original Implementation:
GeoCoordinate PrevLocn = Get();
Sleep(5);
GeoCoordinate Location = Get();
double Dist = Distance(PrevLocn, Location);
double Speed = Dist / 5;
Print(Speed);

New Implementation:
Uncertain<GeoCoordinate> PrevLocn = Get();
Sleep(5);
Uncertain<GeoCoordinate> Location = Get();
Uncertain<double> Dist = Distance(PrevLocn, Location);
Uncertain<double> Speed = Dist / 5;
Print(Speed); // Expected value at 95 CI

Insight: By putting estimates into context and adding domain knowledge, we can improve estimate accuracy.

Estimate: GPS location reading
Context: Car on the road

Estimate: Sleep status
Context: User profiles like age

Estimate: Alzheimer disease diagnosis
Context: User profiles and medical history

Trifest: Two program constructs and runtime support

\[
P(Y | X, Z) = \frac{P(Y | X) P(Z | Y)}{P(Z | X)} \quad (\text{assume } P(Z | X, Y) = P(Z | Y))
\]

\(<|\): To specify domain knowledge \(P(Z | Y)\)

\(#\): To compose domain knowledge and estimates

Sampling at runtime for lazy evaluation

\[
\text{if } (\text{Speed} > 4) \\
\quad \text{Alert("Keep it up!"});
\]