iterata
MEDICAL SERVICES

Two Speed IT

The Lambda Architecture in a Nutshell

Lambda-Architecture-Nutshell.indd
VERSION 20150701-1017

iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

Batch

The needs of the system outweigh the needs of individual events and queries running in flight or active within the system

iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

Increment

The needs of the individual event or query outweigh the needs of the aggregate events or queries in flight in the system

Speed layer

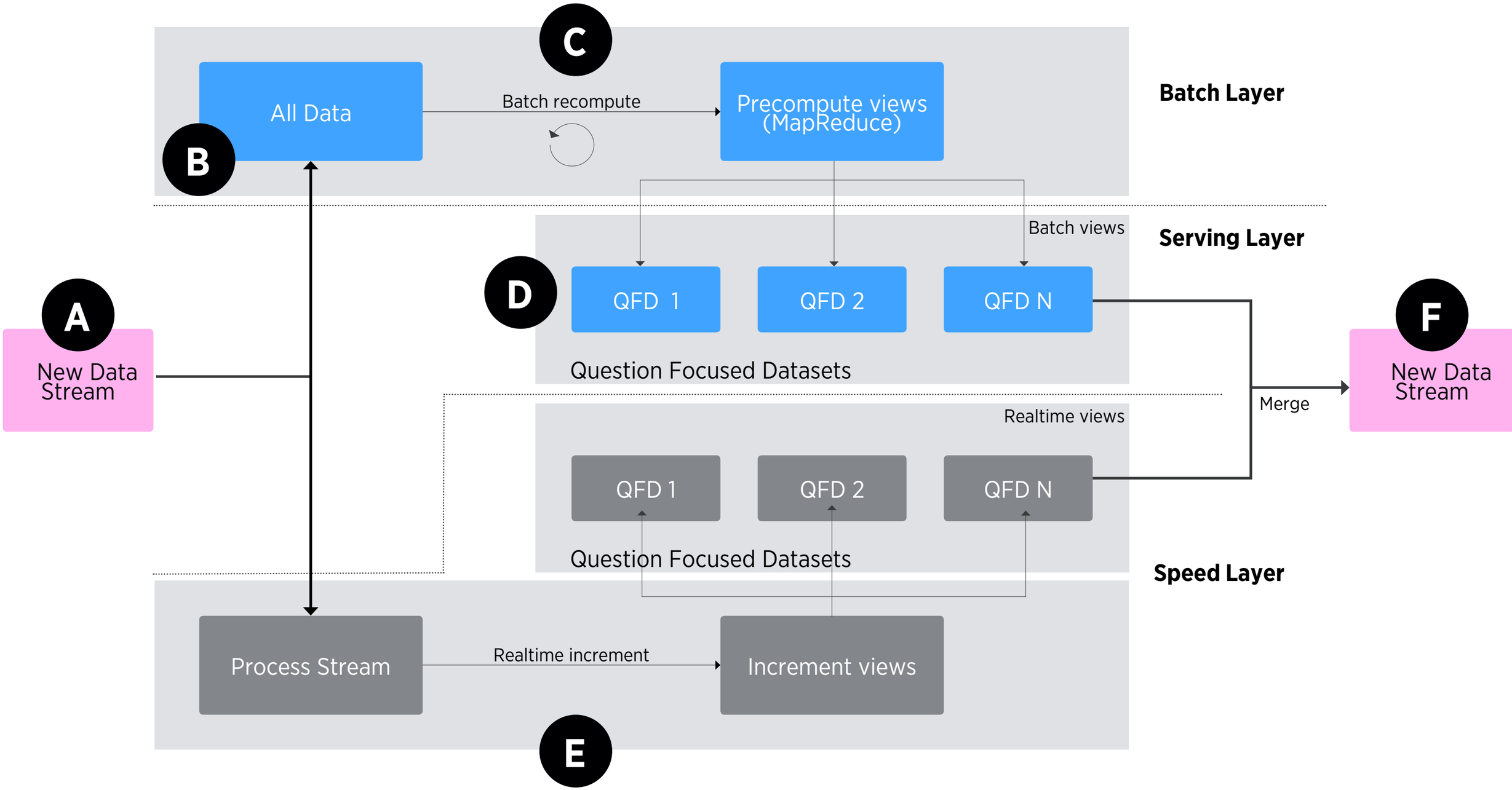
Query = Merging speed and batch layer

Lambda Architecture

iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

Lambda Architecture

iterata
MEDICAL SERVICES



iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

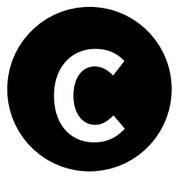
Lambda: A

All new data is sent to both the batch layer and the speed layer. In the batch layer, new data is appended to the master dataset. In the speed layer, the new data is consumed to do incremental updates of the realtime views.

Lambda: B

The master dataset is an immutable, append-only set of data. The master dataset only contains the **rawest** information that is **not derived** from any other information you have.

iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

Lambda: 

The batch layer **precomputes** query functions from scratch. The results of the batch layer are called **batch views**. The batch layer runs in a while(true) loop and continuously recomputes the batch views from scratch. **The strength of the batch layer is its ability to compute arbitrary functions on arbitrary data.** This gives it the power to support any application.

Lambda: D

The serving layer indexes the batch views produced by the batch layer and makes it possible to get particular values out of a batch view very quickly.

The serving layer is a scalable database that swaps in new batch views as they're made available.

Because of the latency of the batch layer, the results available from the serving layer are always **out of date by a few hours.**

Lambda: E

The speed layer compensates for the high latency of updates to the serving layer. It uses fast incremental algorithms and read/write databases to produce realtime views that are always up to date. The speed layer only deals with recent data, because any data older than that has been absorbed into the batch layer and accounted for in the serving layer.
The speed layer is significantly more complex than the batch and serving layers, but that complexity is compensated by the fact that the realtime views can be continuously discarded as data makes its way through the batch and serving layers. So, the potential negative impact of that complexity is greatly limited.

Lambda: 

Queries are resolved by getting results from both the batch and realtime views and **merging** them together.

iterata AG Industriestrasse 8 5722 Gränichen Schweiz info@iterata.ch Tel. +41 62 842 88 27 Fax +41 62 842 88 28

Lambda: Batch View

- **Precomputed** Queries are central to Complex Event Processing / Event Stream Processing architectures.
- Unfortunately, though, most DBMS's still offer only synchronous blocking RPC access to underlying data when asynchronous guaranteed delivery would be preferable for view construction leveraging CEP/ESP techniques.

Lambda: Merging...

- Possibly one of the most difficult aspects of near real-time and historical data integration is combining flows sensibly.
- For example, is the order of interleaving across merge sources applied in a known deterministically recomputable order? If not, how can results be recomputed subsequently? Will data converge?

[cf: http://cs.brown.edu/research/aurora/hwang.icde05.ha.pdf]