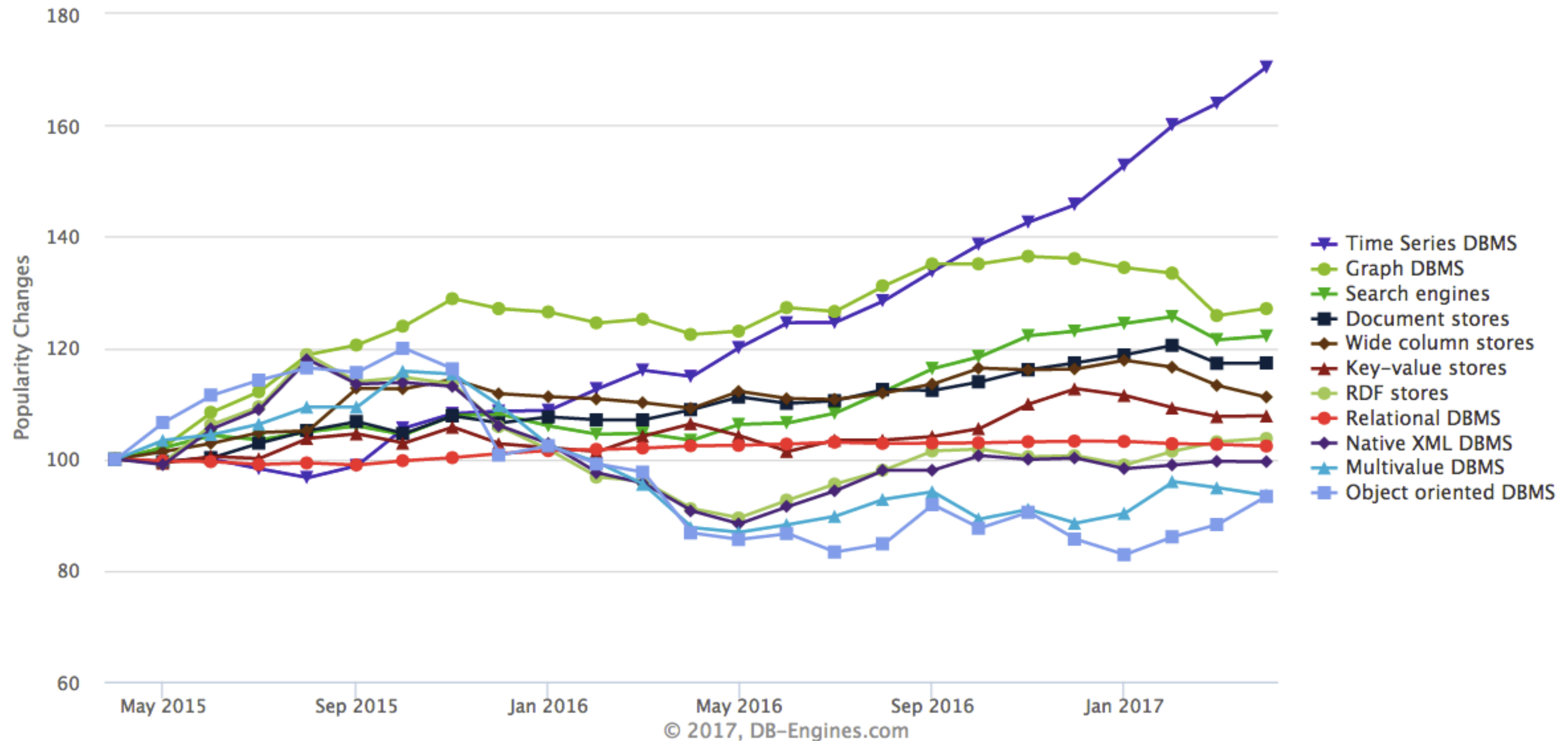


What the heck is time-series data (and why do I need a time-series database?)

Fastest growing database category



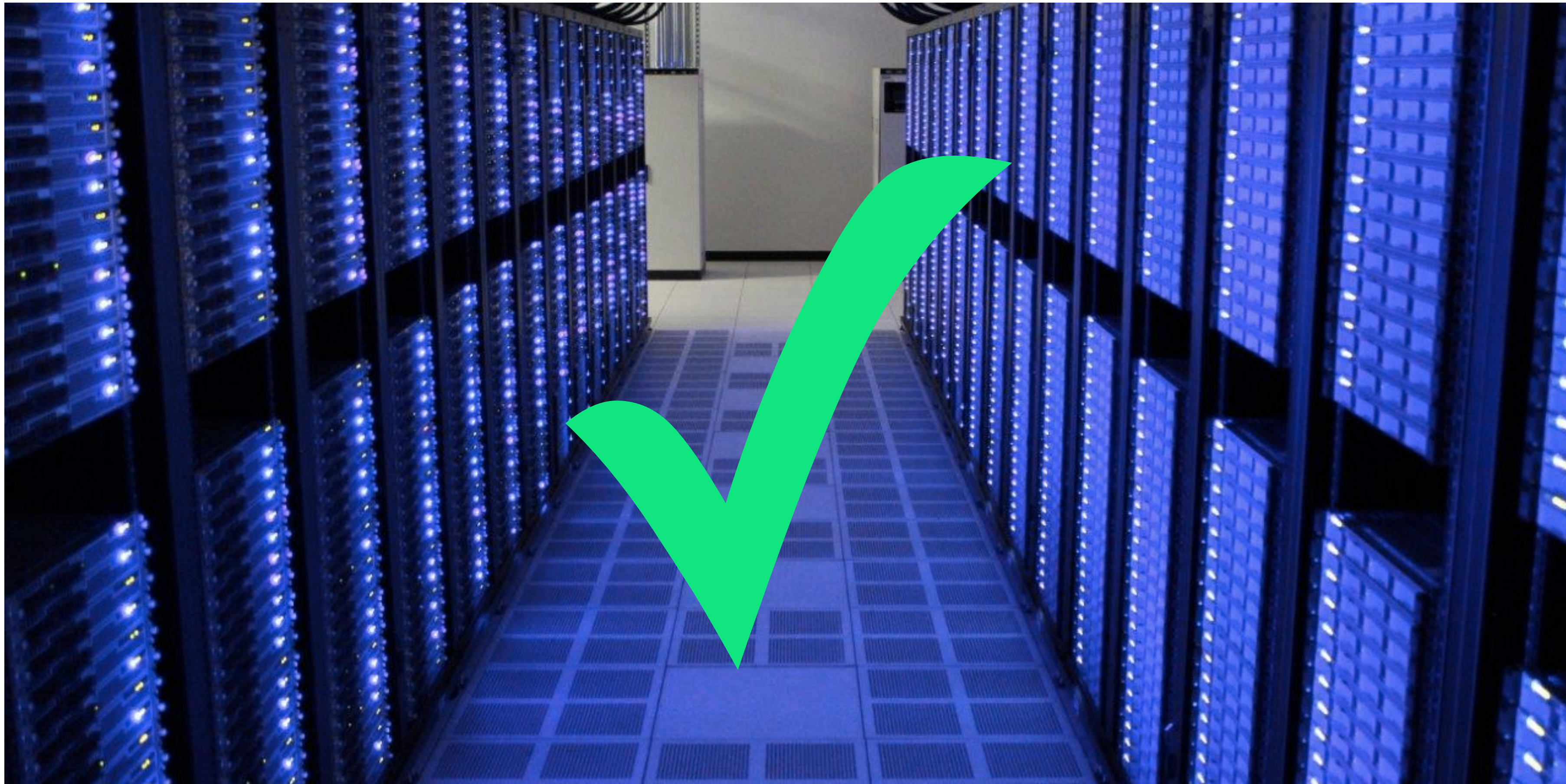
In this talk

1. What is **time-series data**?
(hint: it's not what you think)
2. Why do I need a **time-series database**?
3. Is this just a fad?

What is **time-series** data?

Q: Metrics and Logging?

CPU, free memory, gc pauses, error reports,
application instrumentation, etc.



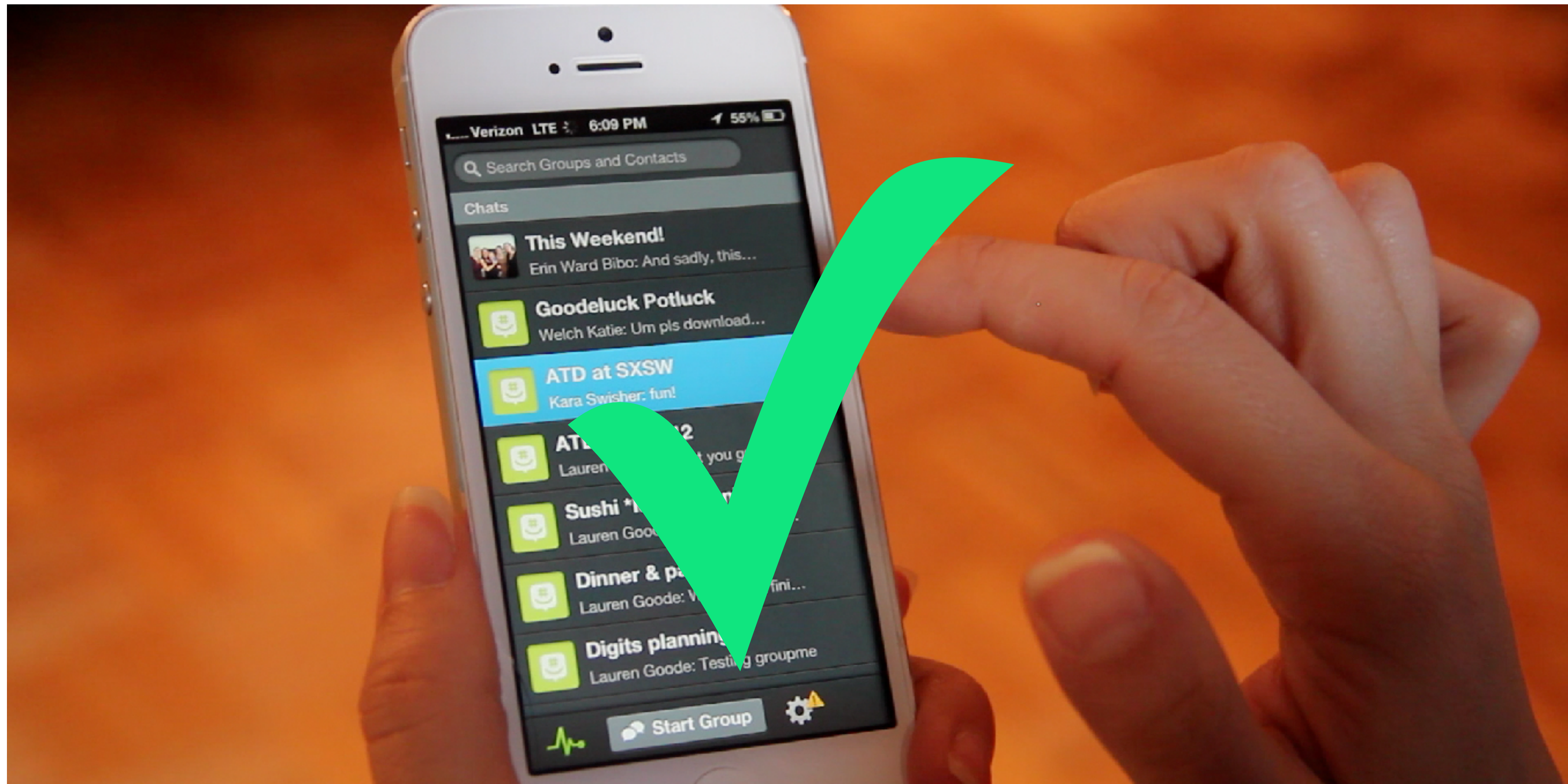
Q: Financial data?

Stock tick stream, payment records,
transaction records



Q: Event data?

Clickstreams, application events,
outages, errors, system status



Q: IoT data?

Sensor data, machine data, industrial monitoring, smart home, wearables



Q: Other data?

Logistics tracking,
environmental monitoring



A: All of the above

So what is **time-series** data?

Time-series data has 3 characteristics

1. Time-centric data

- Capturing and analyzing measurements/events over time.

2. Primarily INSERTS

- Workloads generally write new data. Rarely update.

3. Writes to recent interval

- Data generally written to most recent time interval (although delays possible).

How is this different than having a time field?
Treat changes as inserts, not overwrites.

You can do more with time-series data

PAST

- Analyze historical trends.
- Look at the state of the system at any point in time.

PRESENT

- Real-time monitoring
- Troubleshoot problems as they occur

FUTURE

- Identify and fix problems before they occur, reducing downtime.

What does **time-series data** look like?

(hint: it's not what you think)

What you have been told

Name CPU

Tags Host=Name,Region=West

Data

1990-01-01	01:02:00	70
1990-01-01	01:03:00	71
1990-01-01	01:04:00	72
1990-01-01	01:04:00	73
1990-01-01	01:04:00	
100		

What you have been told

<i>Name</i>	CPU	FreeMem
<i>Tags</i>	Host=Name,Region=West	Host=Name,Region=West
<i>Data</i>	1990-01-01 01:02:00 70 1990-01-01 01:03:00 71 1990-01-01 01:04:00 72 1990-01-01 01:04:00 72 1990-01-01 01:04:00 72 1990-01-01 01:04:00 72 100	1990-01-01 01:02:00 800M 1990-01-01 01:03:00 600M 1990-01-01 01:04:00 400M 1990-01-01 01:04:00 200M 1990-01-01 01:04:00 0

2 time-series?

This is wrong

Time-series data has a richer structure

Tags	Host=Name,Region=Wes			
	t	CPU	MemFree	Temp
Data		70	800M	80
	1990-01-01 01:02:00	71	600M	81
	1990-01-01 01:03:00	72	400M	82
	1990-01-01 01:04:00	73	200M	83
	1990-01-01 01:04:00	100	0	120
	1990-01-01 01:04:00			

Fewer queries

Tags Host=Name,Region=Wes

t

CPU

MemFree

Temp

Data

70

800M

80

1990-01-01 01:02:00

71

600M

81

1990-01-01 01:03:00

72

400M

82

1990-01-01 01:04:00

73

200M

83

1990-01-01 01:04:00

100

0

120

1990-01-01 01:04:00

select * where time = x

Complex filters

<i>Tags</i>	Host=Name,Region=Wes			CPU	MemFree	Temp
	t					
<i>Data</i>				70	800M	80
	1990-01-01	01:02:00		71	600M	81
	1990-01-01	01:03:00		72	400M	82
	1990-01-01	01:04:00		73	200M	83
	1990-01-01	01:04:00		100	0	120
	1990-01-01	01:04:00				

where temp > 100

Complex aggregates

Tags Host=Name,Region=Wes

t

CPU **MemFree** **Temp**

Data

		70	800M	80
1990-01-01	01:02:00	71	600M	81
1990-01-01	01:03:00	72	400M	82
1990-01-01	01:04:00	73	200M	83
1990-01-01	01:04:00	100	0	120
1990-01-01	01:04:00			

avg(mem_free) group by (cpu/10)

Correlations

Tags Host=Name,Region=Wes

t

CPU ↔ MemFree ↔ Temp

Data

	70	800M	80
1990-01-01 01:02:00	71	600M	81
1990-01-01 01:03:00	72	400M	82
1990-01-01 01:04:00	73	200M	83
1990-01-01 01:04:00	100	0	120
1990-01-01 01:04:00			

how does temperature correlate with mem_free?

Leverage relations

<i>Data</i>			CPU	Host	Region
	1990-01-01	01:02:00	70	1	91
	1990-01-01	01:03:00	71	2	92
	1990-01-01	01:04:00	72	3	93
	1990-01-01	01:04:00	73	4	94
	1990-01-01	01:04:00	100	5	95

Region stored in separate host metadata table

How to **store time-series data**

Can't I use a “normal” database?

You can, and some people do



Source: Percona

TIME SCALE

Golden age of time-series databases



TIMESCALE



Why do I need a specialized
time-series database?

Problem: Time-series data piles up very quickly

25GB data collected per hour by
connected cars (McKinsey)

*“Our Boeing 787s generate half
a terabyte of data per flight”*

- Virgin Atlantic IT director

Time-series databases introduce efficiencies
by treating time as a first-class citizen.

OLTP

- ✗ Primarily UPDATES
- ✗ Writes randomly distributed
- ✗ Transactions to multiple primary keys

Time Series

- ✓ Primarily INSERTs
- ✓ Writes to recent time interval
- ✓ Writes associated with a timestamp and primary key

Time-series databases introduce efficiencies

1. **Better write rates** to handle ingest scale.
2. **Query performance**, even at scale.
3. **Ease of use** via common functions (e.g., interpolation)

Is this just a fad? (No.)

Why time-series databases will continue to be popular

Operational needs

- Managing increasingly complex systems requires:

real-time monitoring,
troubleshooting,
better prediction.

Business needs

- Constant need to make better data-driven decision faster.
- More sources of data: new devices, old devices coming online, new systems.

Tech trends

- Cheaper storage, faster processors, more bandwidth
- Better resources: cloud computing, data analysis tools

Crazy idea: Is all data time-series data?

TIMESCALE

<https://github.com/timescale/timescaledb>